

# 데이터 분석을 위한 WEB 이해



2023.01

# CH01 – WEB 개발도구

# WEB 개발도구

## ◆ 아나콘다(Anaconda)

- 수 백 개의 파이썬 패키지를 포함하고 있는 파이썬 배포판
- 패키지를 별도로 시간 들여 추가 설치하는 수고 감소
- 기능
  - 패키지 설치 및 관리
  - 가상환경 관리
- 관리 프로그램
  - **conda** 제공



# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ conda 프로그램

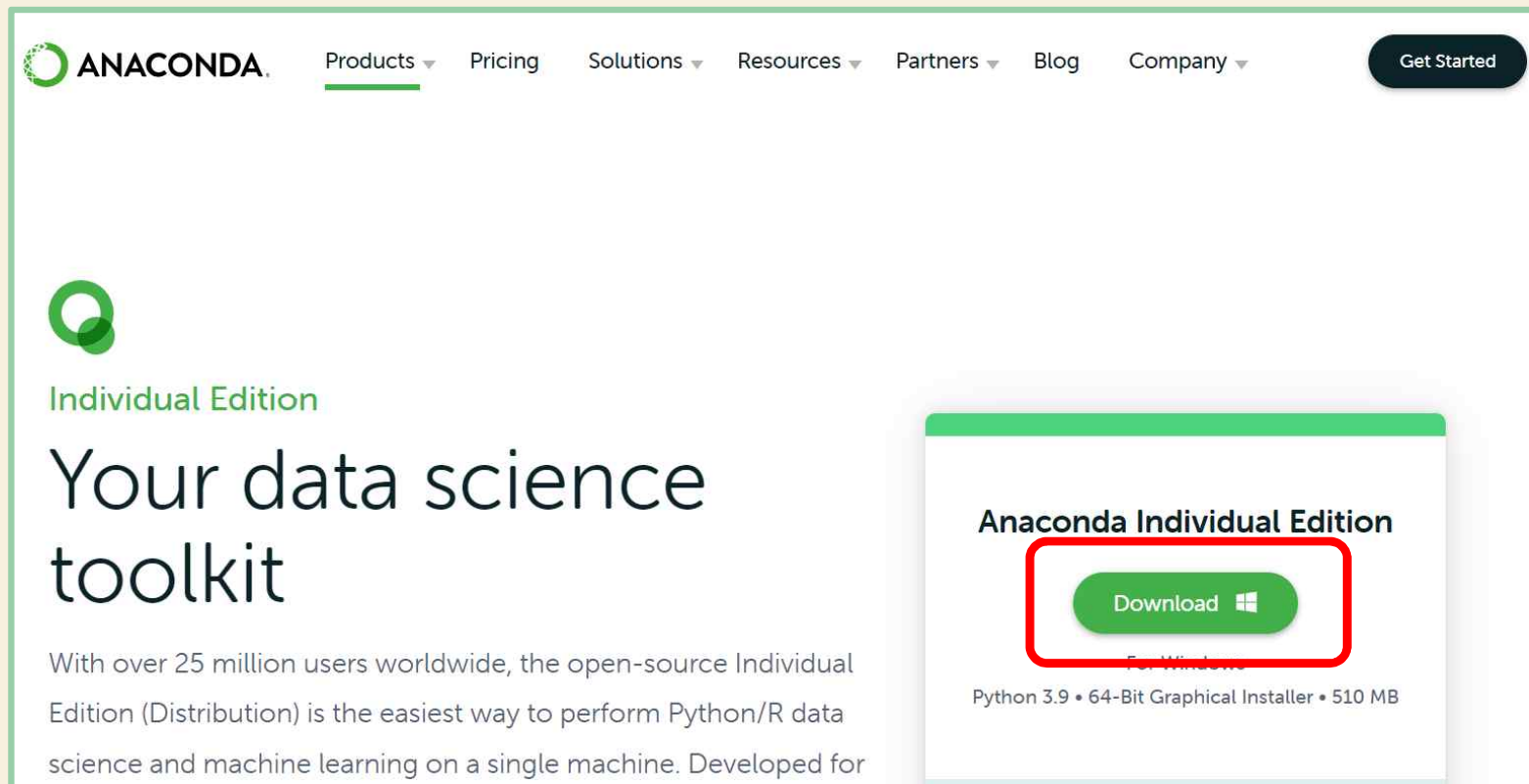
- 자동 의존성 체크 후 설치
- 모든 라이브러리 없으므로 pip와 병행 사용

기능	명령어 옵션 커맨드
패키지 설치	\$ <b>conda install</b> numpy # numpy 설치
동시에 여러 패키지 설치	\$ <b>conda install</b> numpy scipy pandas # numpy, scipy, pandas 동시 설치
특정 버전 설치	\$ <b>conda install</b> numpy=1.10 # 특정 버전 설치
패키지 제거	\$ <b>conda remove</b> package_name
패키지 업데이트	\$ <b>conda update</b> package_name
모든 패키지 업데이트	\$ <b>conda update</b> -all
설치된 목록 출력	\$ <b>conda list</b> # 정보 확인
설치하려는 패키지 검색	\$ <b>conda search</b> search_term

# WEB 개발도구

## ◆ 아나콘다(Anaconda)

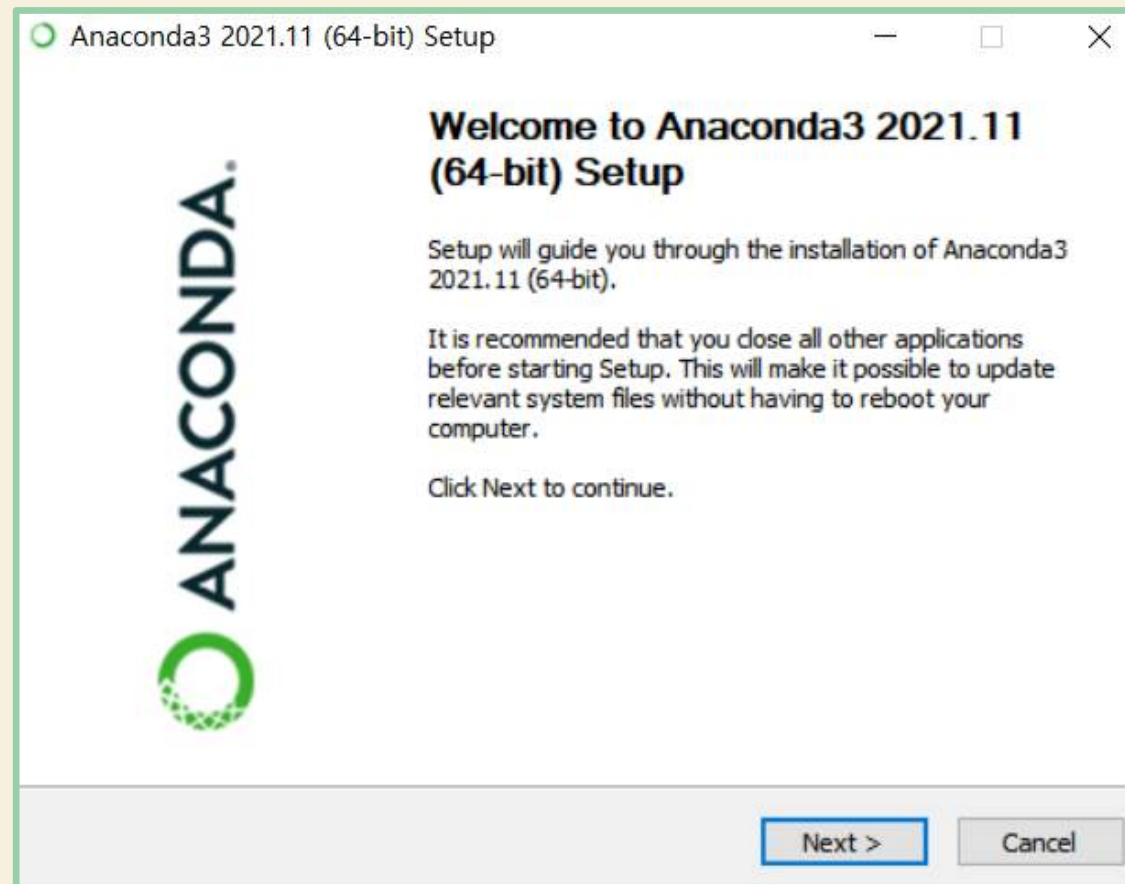
➤ Anaconda3-2021.11-Windows-x86\_64.exe 다운로드



# WEB 개발도구

## ◆ 아나콘다(Anaconda)

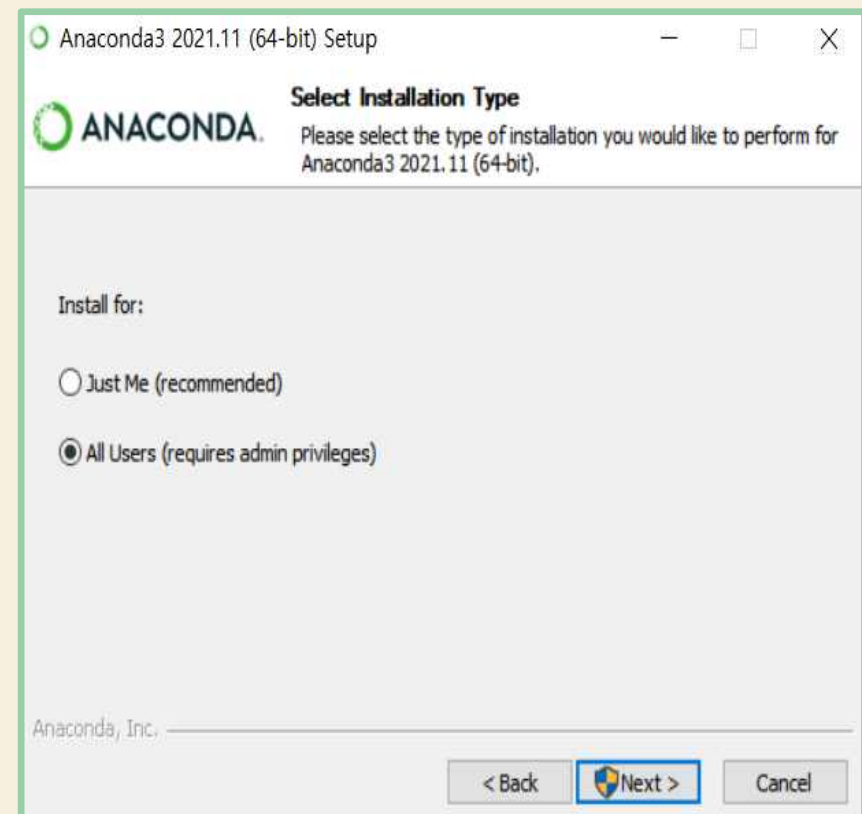
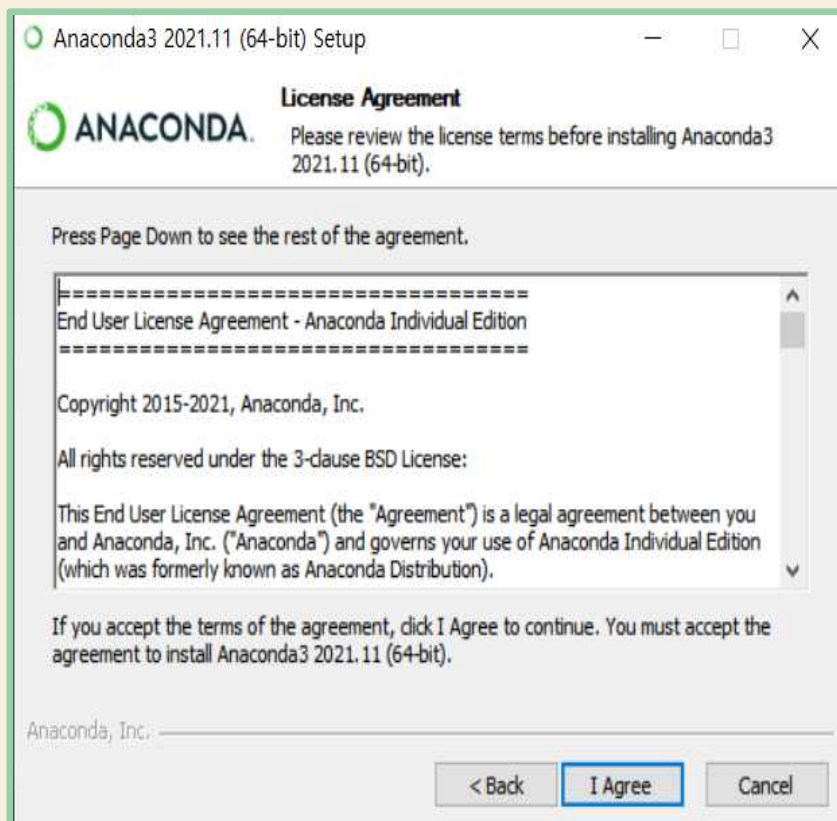
- Anaconda3-2021.11-Windows-x86\_64.exe 실행



# WEB 개발도구

## ◆ 아나콘다(Anaconda)

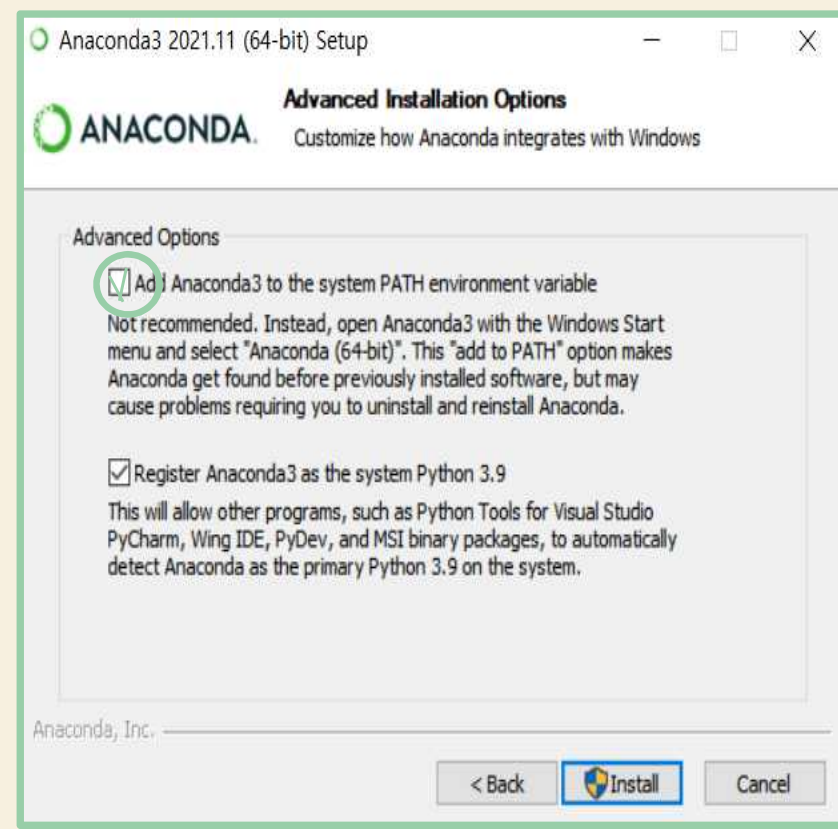
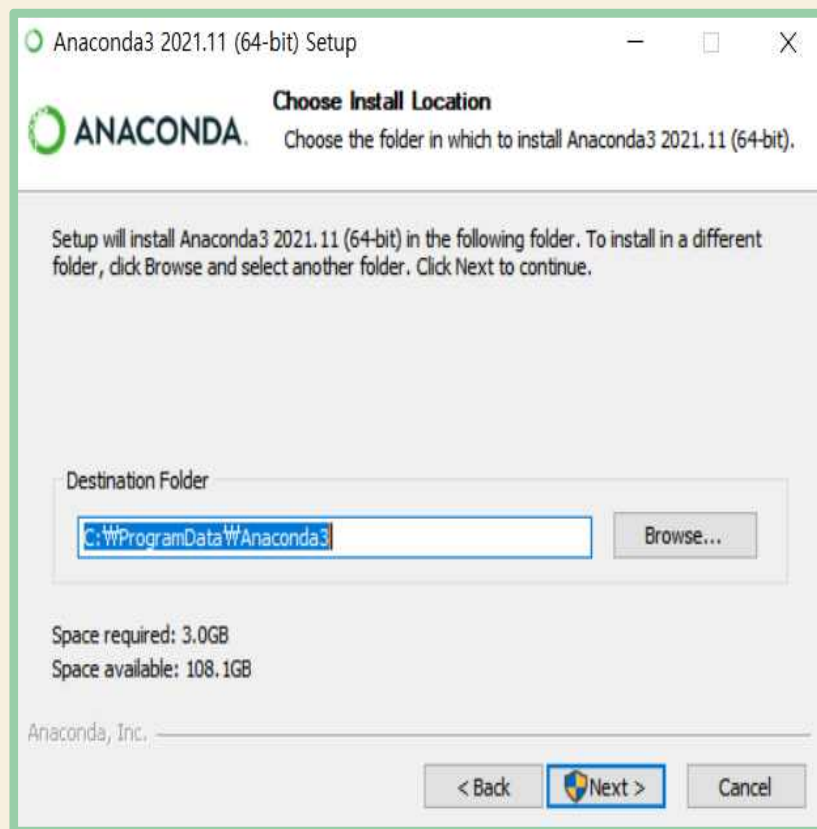
➤ Anaconda3-2021.11-Windows-x86\_64.exe 실행



# WEB 개발도구

## ◆ 아나콘다(Anaconda)

➤ Anaconda3-2021.11-Windows-x86\_64.exe 실행

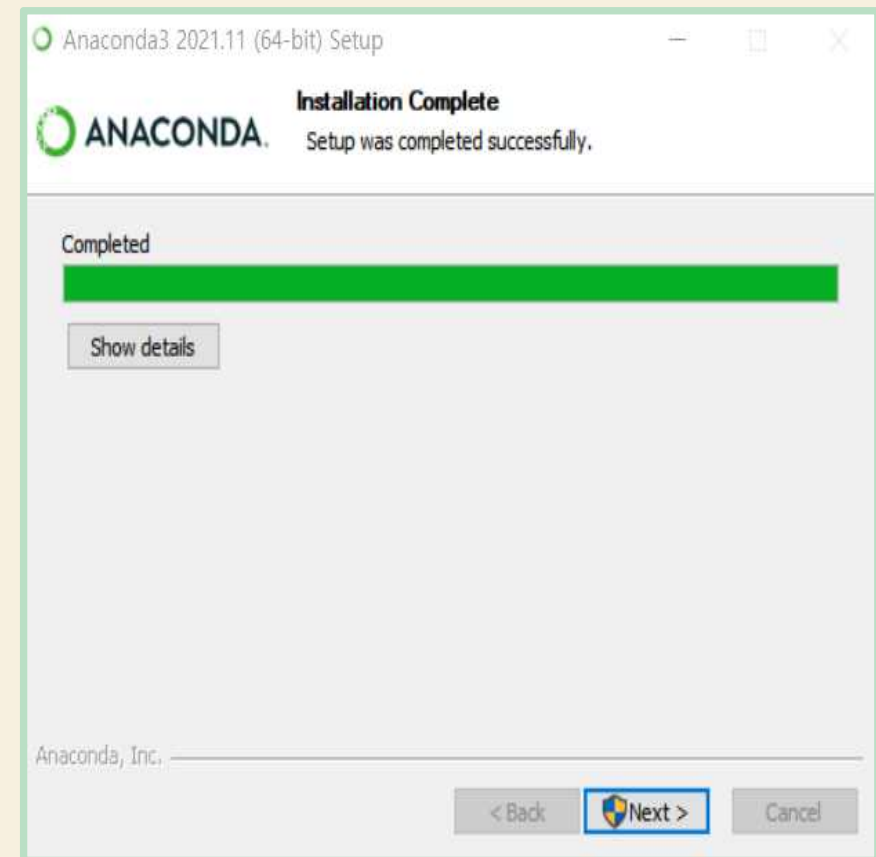
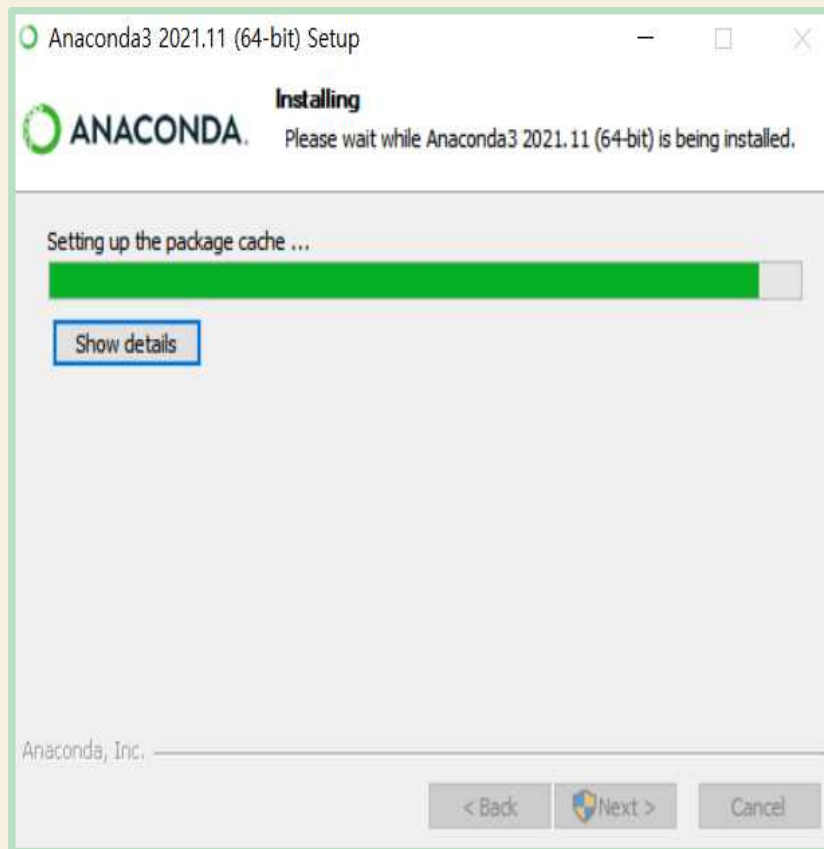




# WEB 개발도구

## ◆ 아나콘다(Anaconda)

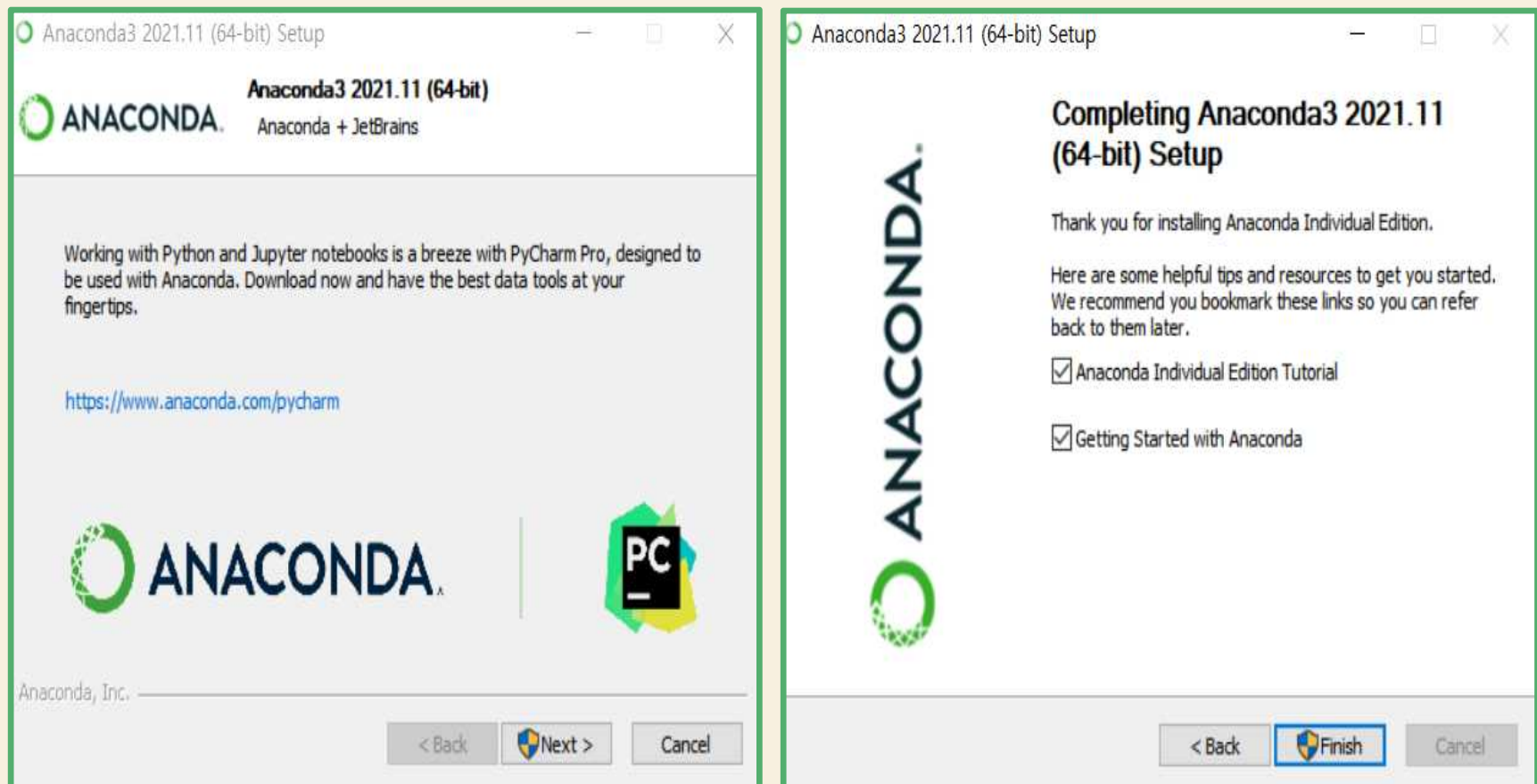
➤ Anaconda3-2021.11-Windows-x86\_64.exe 실행



# WEB 개발도구

## ◆ 아나콘다(Anaconda)

➤ Anaconda3-2021.11-Windows-x86\_64.exe 실행

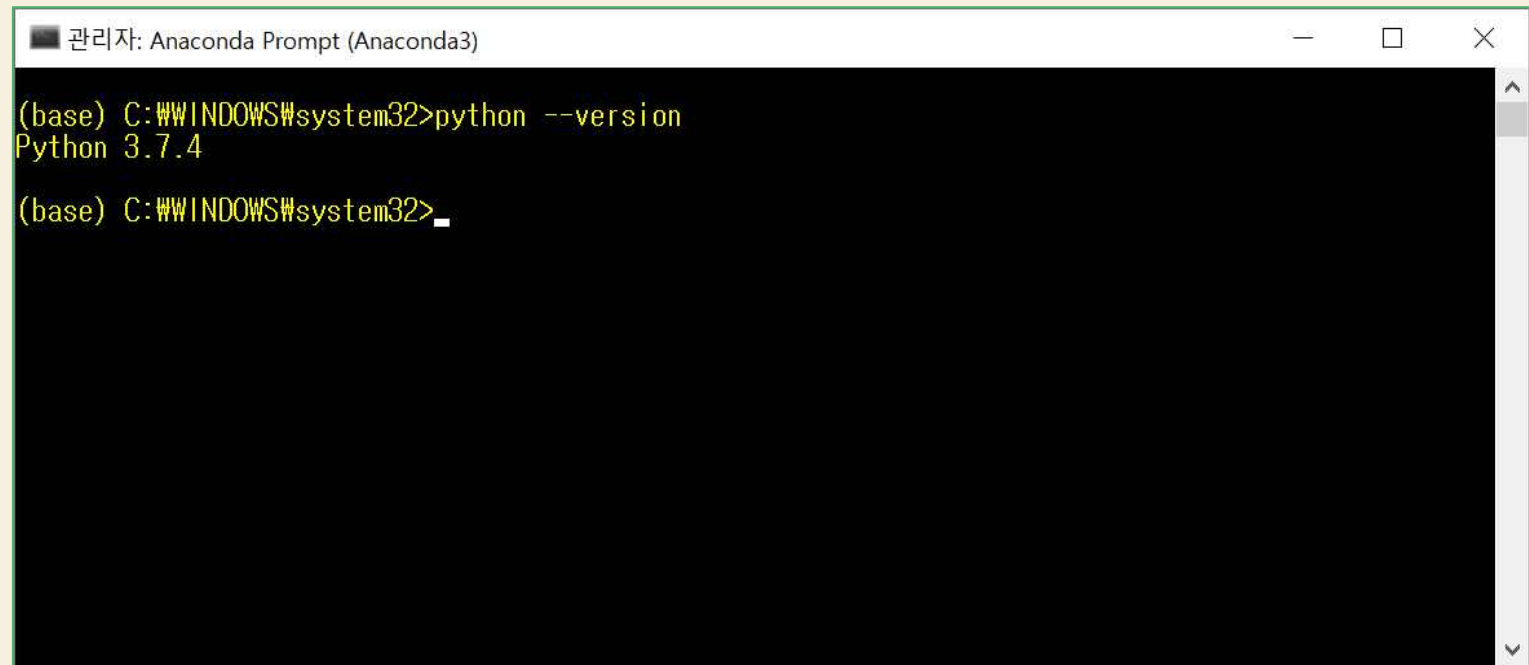


# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ Python 설치 버전 확인

- Anaconda Prompt 관리자 권한으로 실행
- `python --version` 입력 => 버전 확인



```
관리자: Anaconda Prompt (Anaconda3)

(base) C:\WINDOWS\system32>python --version
Python 3.7.4

(base) C:\WINDOWS\system32>_
```

# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ 업데이트

- Conda 업데이트 : `conda update -n base conda`

```
관리자: Anaconda Prompt (Anaconda3) - conda update -n base conda

(base) C:\WINDOWS\system32>conda update -n base conda
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\hanece\Anaconda3

added / updated specs:
- conda

The following packages will be downloaded:

package                                     build                                py_0    2.8 MB    11 KB
backports.functools_lru_cache-1.6.1|      py37_1
conda-4.8.0                                py37_0    656 KB
future-0.18.2                               |
-----
Total:                                     3.5 MB

The following packages will be UPDATED:

backports.functoo~      1.5-py_2 --> 1.6.1-py_0
conda                   4.7.12-py37_0 --> 4.8.0-py37_1
future                 0.17.1-py37_0 --> 0.18.2-py37_0

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

# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ 업데이트

- Conda 패키지 업데이트 : **conda update --all**

The first screenshot shows the command `conda update --all` being executed. It displays the process of collecting metadata, solving the environment, and showing a warning about two possible package resolutions. It then shows the package plan and the environment location.

```
done
(base) C:\WINDOWS\system32>conda update --all
Collecting package metadata (current_repodata.json): done
Solving environment: /
Warning: 2 possible package resolutions (only showing highest):
- defaults::jupyter_console-5.2.0-py37_1, default
- defaults::jupyter_console-6.0.0-py37_0, default

## Package Plan ##

environment location: C:\Users\wance\Anaconda3

The following packages will be downloaded:

package                                     build
-----
_anaconda_depends-2019.03                  py37_0
anaconda-custom                             py37_1
anaconda-project-0.8.4                     py_0
```

The second screenshot shows the list of packages to be downgraded:

```
The following packages will be DOWNGRADED:
anaconda                2019.10-py37_0 --> custom-py37_1
jedi                    0.15.1-py37_0 --> 0.14.1-py37_0
jupyter_console         6.0.0-py37_0 --> 5.2.0-py37_1
pycosat                 0.6.3-py37hfaf6e2cd_0 --> 0.6.3-py37he774522_0

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

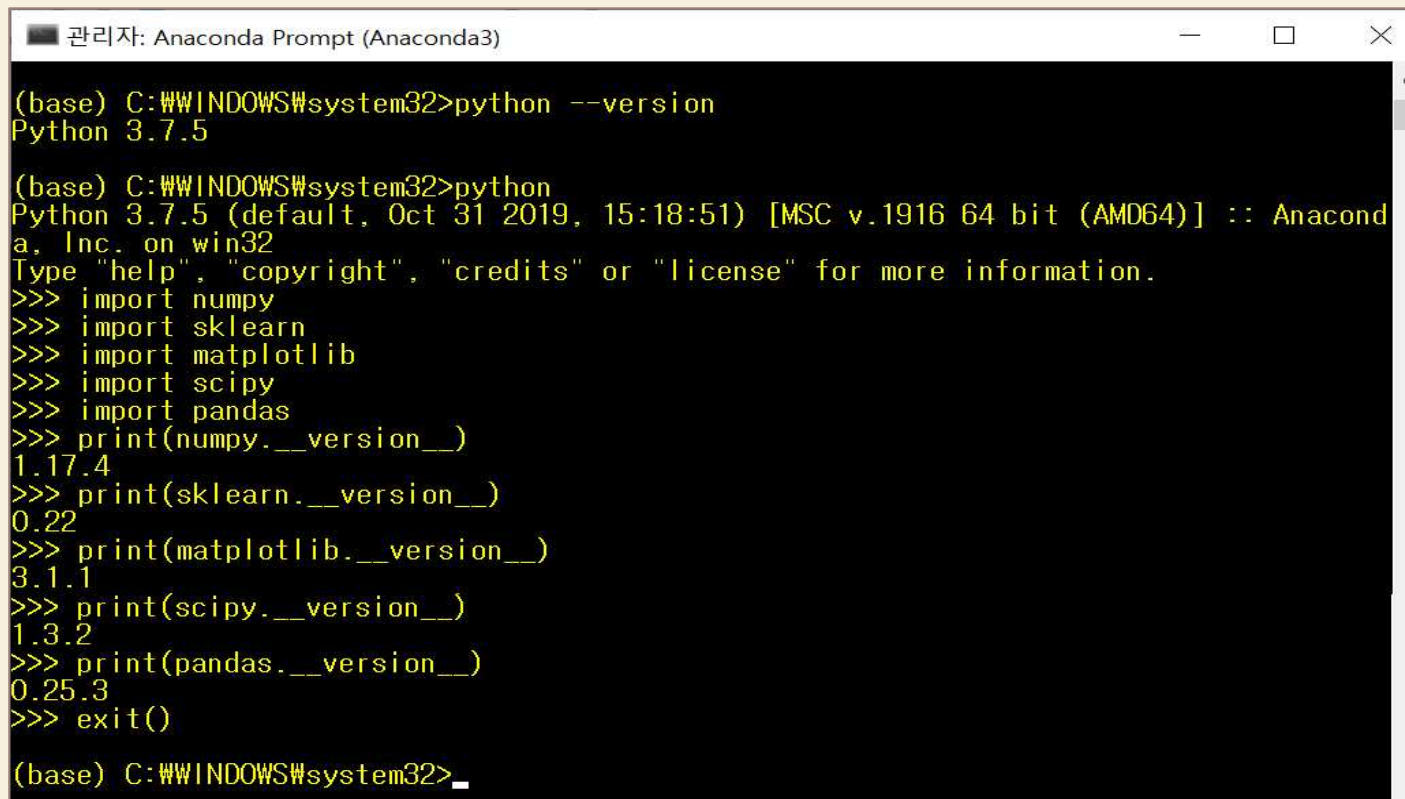
The third screenshot shows the final output of the update process, including the installation of the new packages and the reset of the Jupyter environment.

```
(base) C:\WINDOWS\system32>conda update --all
done
(base) C:\WINDOWS\system32>
```

# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ 업데이트 후 설치 라이브러리 확인

A screenshot of the Anaconda Prompt terminal window. The window title is '관리자: Anaconda Prompt (Anaconda3)'. The terminal shows the following commands and output:

```
(base) C:\WINDOWS\system32>python --version
Python 3.7.5

(base) C:\WINDOWS\system32>python
Python 3.7.5 (default, Oct 31 2019, 15:18:51) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import numpy
>>> import sklearn
>>> import matplotlib
>>> import scipy
>>> import pandas
>>> print(numpy.__version__)
1.17.4
>>> print(sklearn.__version__)
0.22
>>> print(matplotlib.__version__)
3.1.1
>>> print(scipy.__version__)
1.3.2
>>> print(pandas.__version__)
0.25.3
>>> exit()

(base) C:\WINDOWS\system32>_
```



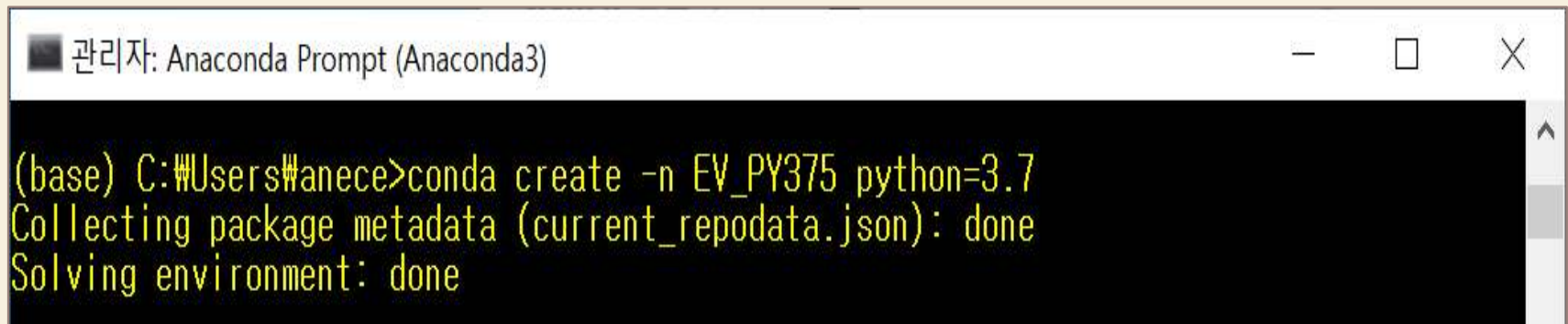
# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ 가상환경 만들기

`conda create -n 가상환경이름 python=파이썬버전`

(base) C:\Users\사용자계정>`conda create -n MYSERVER python=3.9`



```
관리자: Anaconda Prompt (Anaconda3)

(base) C:\Users\anece>conda create -n EV_PY375 python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ 가상환경 만들기

```
## Package Plan ##
environment location: C:\Users\wanece\Anaconda3\envs\EV_PY375
added / updated specs:
- python=3.7

The following packages will be downloaded:

package | build |
-----|-----|
wheel-0.33.6 | py37_0 | 58 KB
-----|-----|
Total: | 58 KB |

The following NEW packages will be INSTALLED:

ca-certificates pkgs/main/win-64::ca-certificates-2019.11.27-0
certifi pkgs/main/win-64::certifi-2019.11.28-py37_0
openssl pkgs/main/win-64::openssl-1.1.1d-he774522_3
pip pkgs/main/win-64::pip-19.3.1-py37_0
python pkgs/main/win-64::python-3.7.5-h8c8aaf0_0
setuptools pkgs/main/win-64::setuptools-42.0.2-py37_0
sqlite pkgs/main/win-64::sqlite-3.30.1-he774522_0
vc pkgs/main/win-64::vc-14.1-h0510ff6_4
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_1
wheel pkgs/main/win-64::wheel-0.33.6-py37_0
wincertstore pkgs/main/win-64::wincertstore-0.2-py37_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
wheel-0.33.6 | 58 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
# $ conda activate EV_PY375
#
# To deactivate an active environment, use
#
# $ conda deactivate
```



# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ 가상환경 확인

```
conda env list
```

```
(base) C:\Users\anece>conda env list
# conda environments:
#
base                  *  C:\Users\anece\Anaconda3
EV_PY375              C:\Users\anece\Anaconda3\envs\EV_PY375

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

# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ 가상환경 실행 & 종료

실행 : `conda activate` 가상환경이름

종료: `conda deactivate`

```
(base) C:\Users\#anece>conda activate EV_PY375
```

```
(EV_PY375) C:\Users\#anece>conda deactivate
```

```
(base) C:\Users\#anece>
```

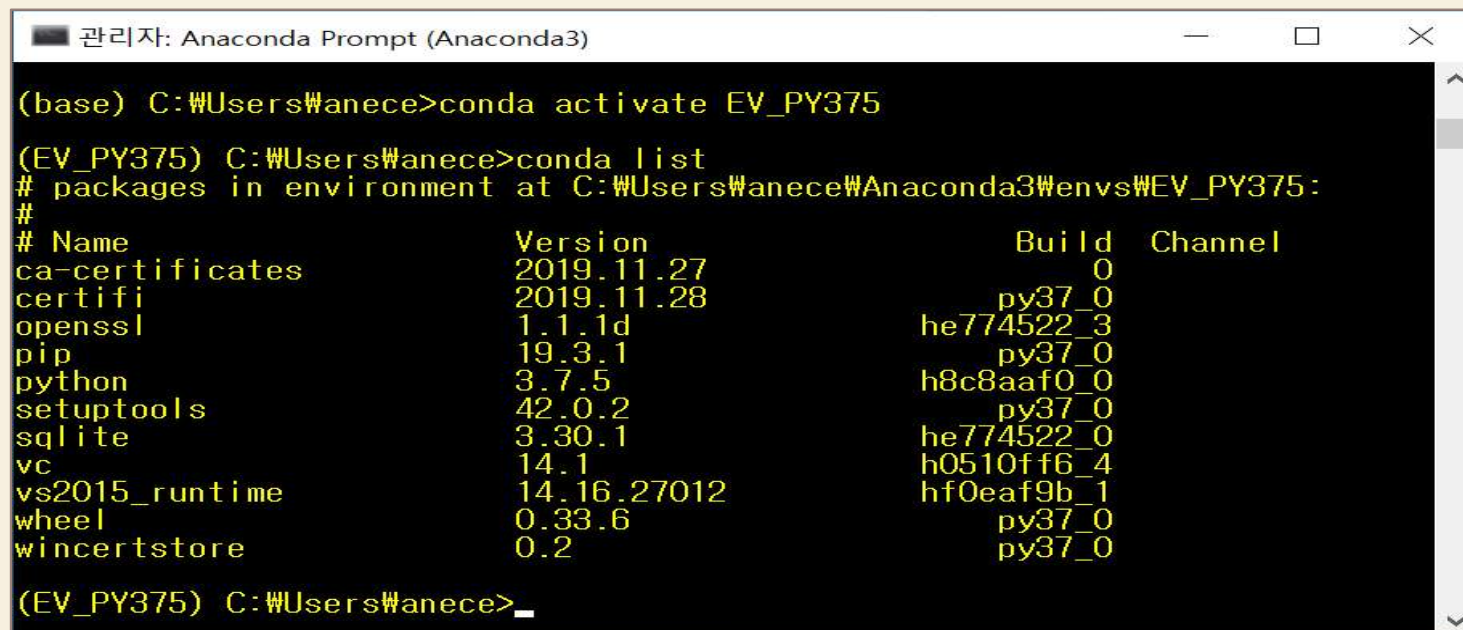
# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ 가상환경 설치 모듈(라이브러리) 체크

설치 모듈 정보 출력 : `conda list`

모듈 설치 명령어 : `conda install 라이브러리명`



```
관리자: Anaconda Prompt (Anaconda3)

(base) C:\Users\Wanece>conda activate EV_PY375

(EV_PY375) C:\Users\Wanece>conda list
# packages in environment at C:\Users\Wanece\Anaconda3\envs\EV_PY375:
#
# Name                        Version      Build      Channel
ca-certificates              2019.11.27   0          py37_0
certifi                      2019.11.28   py37_0
openssl                      1.1.1d       he774522_3 py37_0
pip                          19.3.1       py37_0
python                      3.7.5        h8c8aaf0_0 py37_0
setuptools                   42.0.2       py37_0
sqlite                      3.30.1       he774522_0 py37_0
vc                          14.1         h0510ff6_4 py37_0
vs2015_runtime               14.16.27012 hf0eaf9b_1 py37_0
wheel                       0.33.6       py37_0
wincertstore                 0.2          py37_0

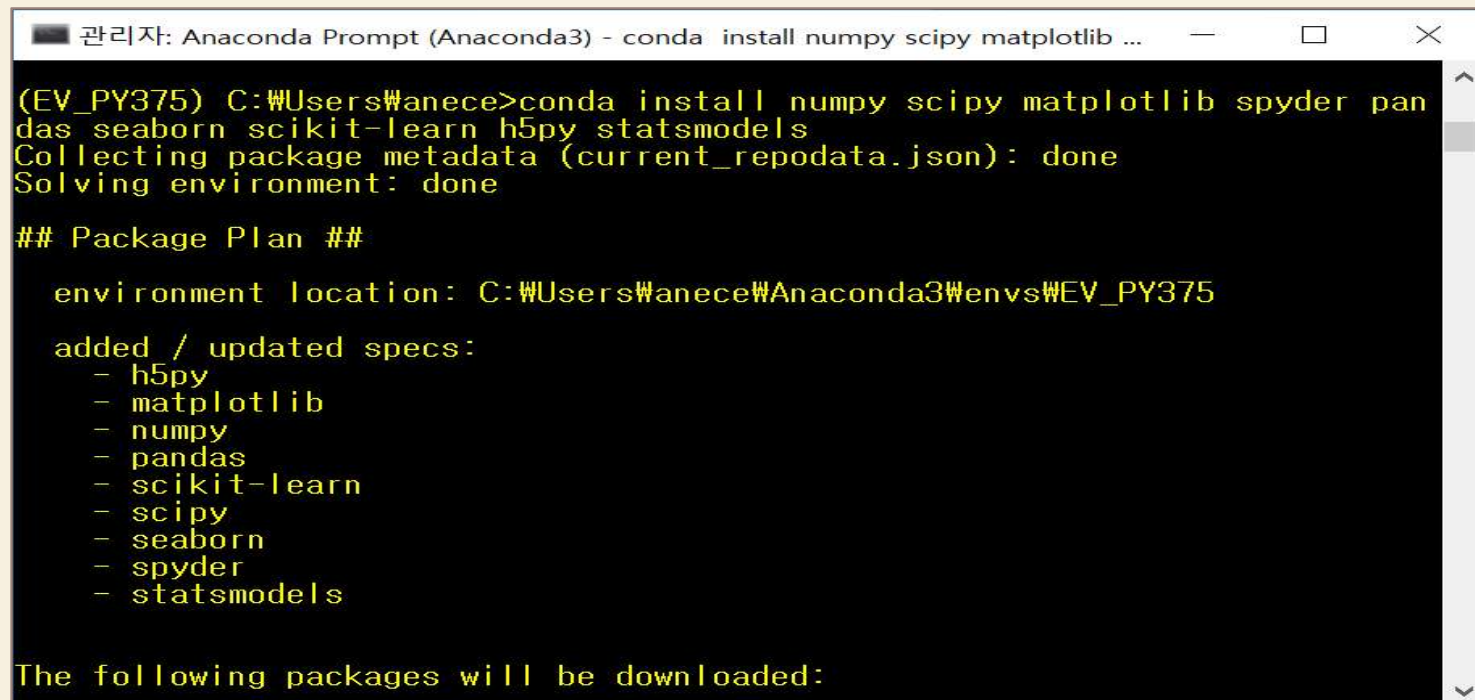
(EV_PY375) C:\Users\Wanece>_
```

# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ 가상환경 설치 모듈(라이브러리) 설치

```
conda install numpy scipy matplotlib spyder  
conda install pandas seaborn scikit-learn  
conda install h5py statsmodels
```



```
관리자: Anaconda Prompt (Anaconda3) - conda install numpy scipy matplotlib ...  
(EV_PY375) C:\Users\Wanece>conda install numpy scipy matplotlib spyder pandas  
seaborn scikit-learn h5py statsmodels  
Collecting package metadata (current_repodata.json): done  
Solving environment: done  
  
## Package Plan ##  
  
environment location: C:\Users\Wanece\Anaconda3\envs\EV_PY375  
  
added / updated specs:  
- h5py  
- matplotlib  
- numpy  
- pandas  
- scikit-learn  
- scipy  
- seaborn  
- spyder  
- statsmodels  
  
The following packages will be downloaded:
```

# WEB 개발도구

## ◆ 아나콘다(Anaconda)

### ➤ 가상환경 설치 모듈(라이브러리) 설치

```
관리자: Anaconda Prompt (Anaconda3) - conda install numpy scipy matplotlib ...
win_inet_pton      pkgs/main/win-64::win_inet_pton-1.1.0-py37_0
wrapt              pkgs/main/win-64::wrapt-1.11.2-py37he774522_0
yaml              pkgs/main/win-64::yaml-0.1.7-hc54c509_2
yapf              pkgs/main/noarch::yapf-0.28.0-py_0
zeromq            pkgs/main/win-64::zeromq-4.3.1-h33f27b4_3
zipp              pkgs/main/noarch::zipp-0.6.0-py_0
zlib              pkgs/main/win-64::zlib-1.2.11-h62dcd97_3

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

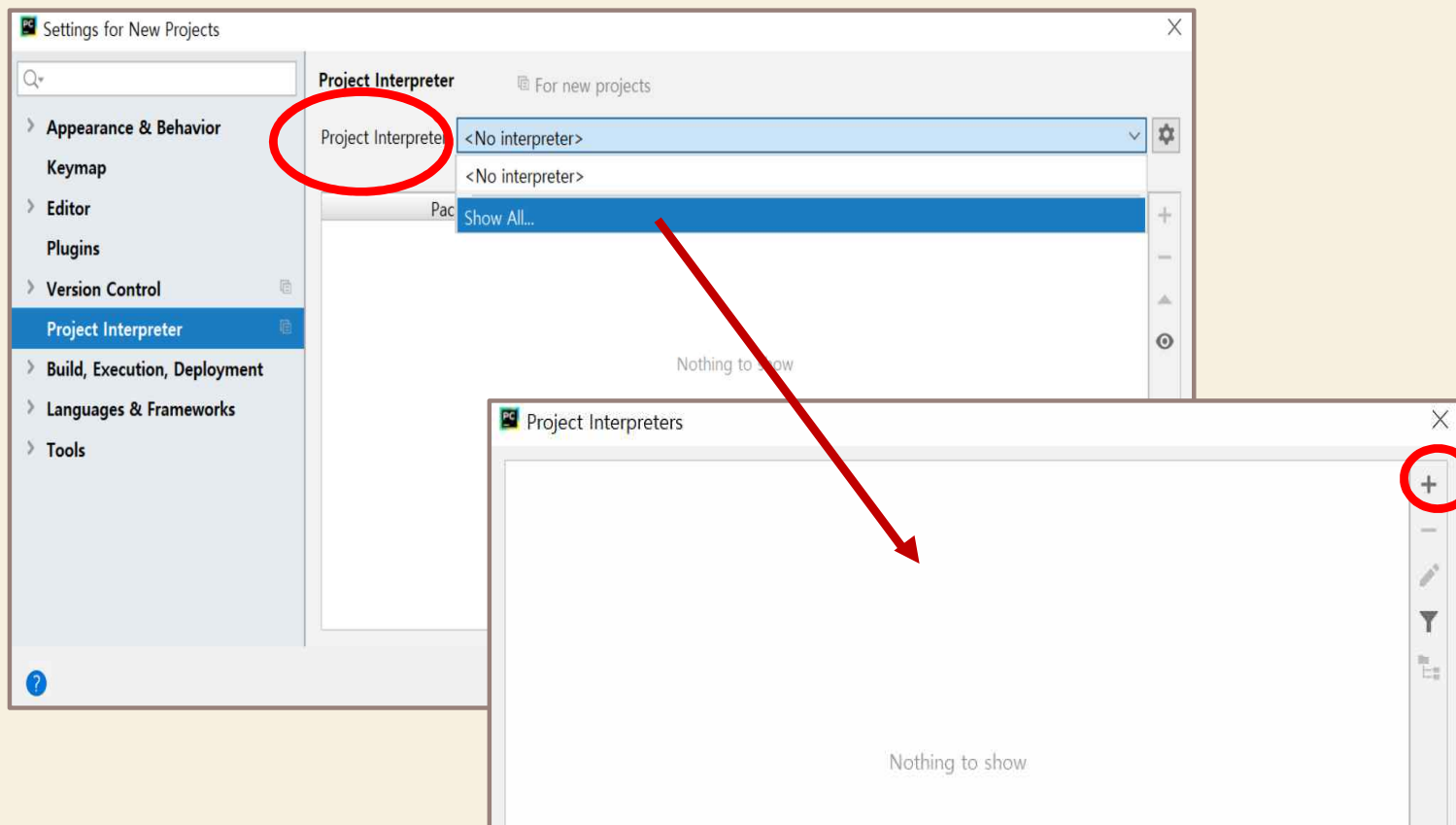
```
관리자: Anaconda Prompt (Anaconda3)
.exe', 'C:\\Users\\anece\\Anaconda3\\envs\\EV_PY375\\Scripts\\spyder-script.py'
, '--reset']
done

(EV_PY375) C:\\Users\\anece>conda list
# packages in environment at C:\\Users\\anece\\Anaconda3\\envs\\EV_PY375:
#
# Name                Version             Build      Channel
alabaster              0.7.12              py37_0
argh                   0.26.2              py37_0
asn1crypto             1.2.0               py37_0
```

# WEB 개발도구

## ◆ 아나콘다(Anaconda) & PyCharm 연동

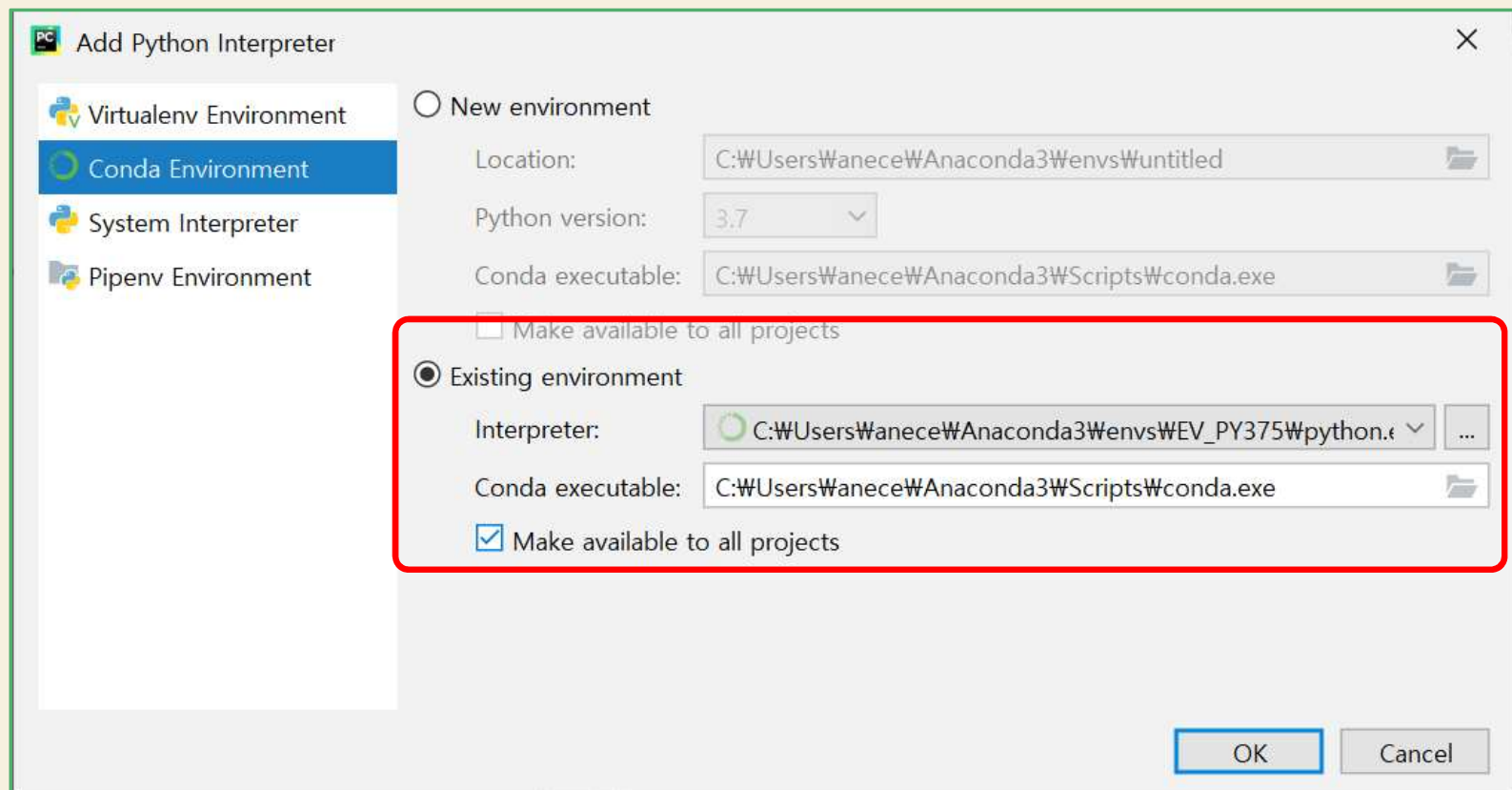
### ➤ Python 인터프리터 설정



# WEB 개발도구

## ◆ 아나콘다(Anaconda) & PyCharm 연동

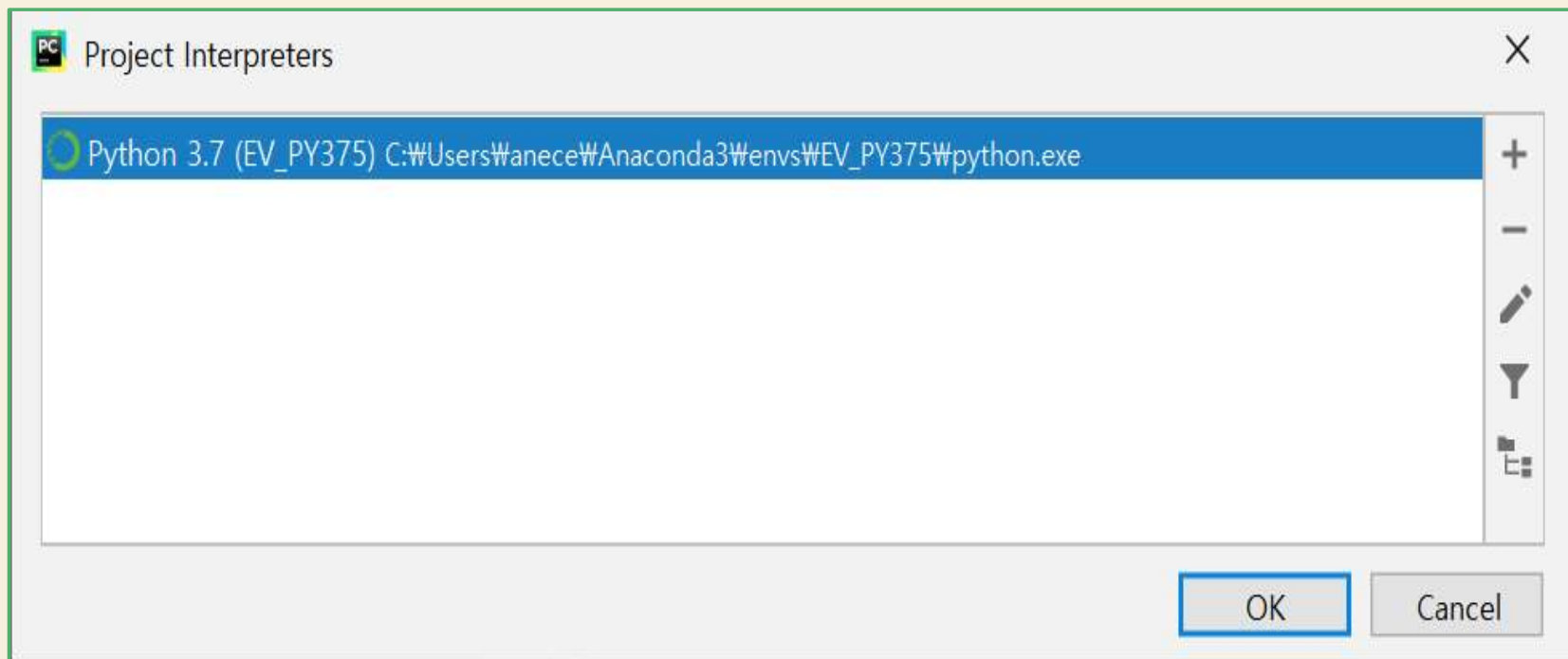
### ➤ Python 인터프리터 설정



# WEB 개발도구

## ◆ 아나콘다(Anaconda) & PyCharm 연동

### ➤ Python 인터프리터 설정

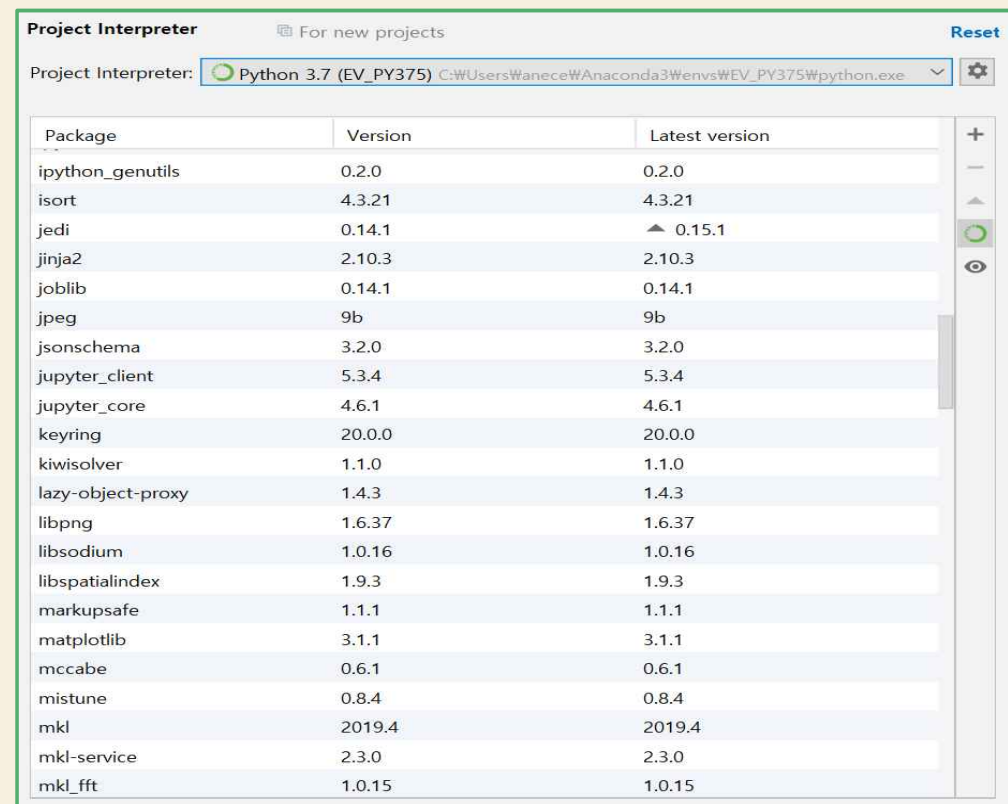




# WEB 개발도구

## ◆ 아나콘다(Anaconda) & PyCharm 연동

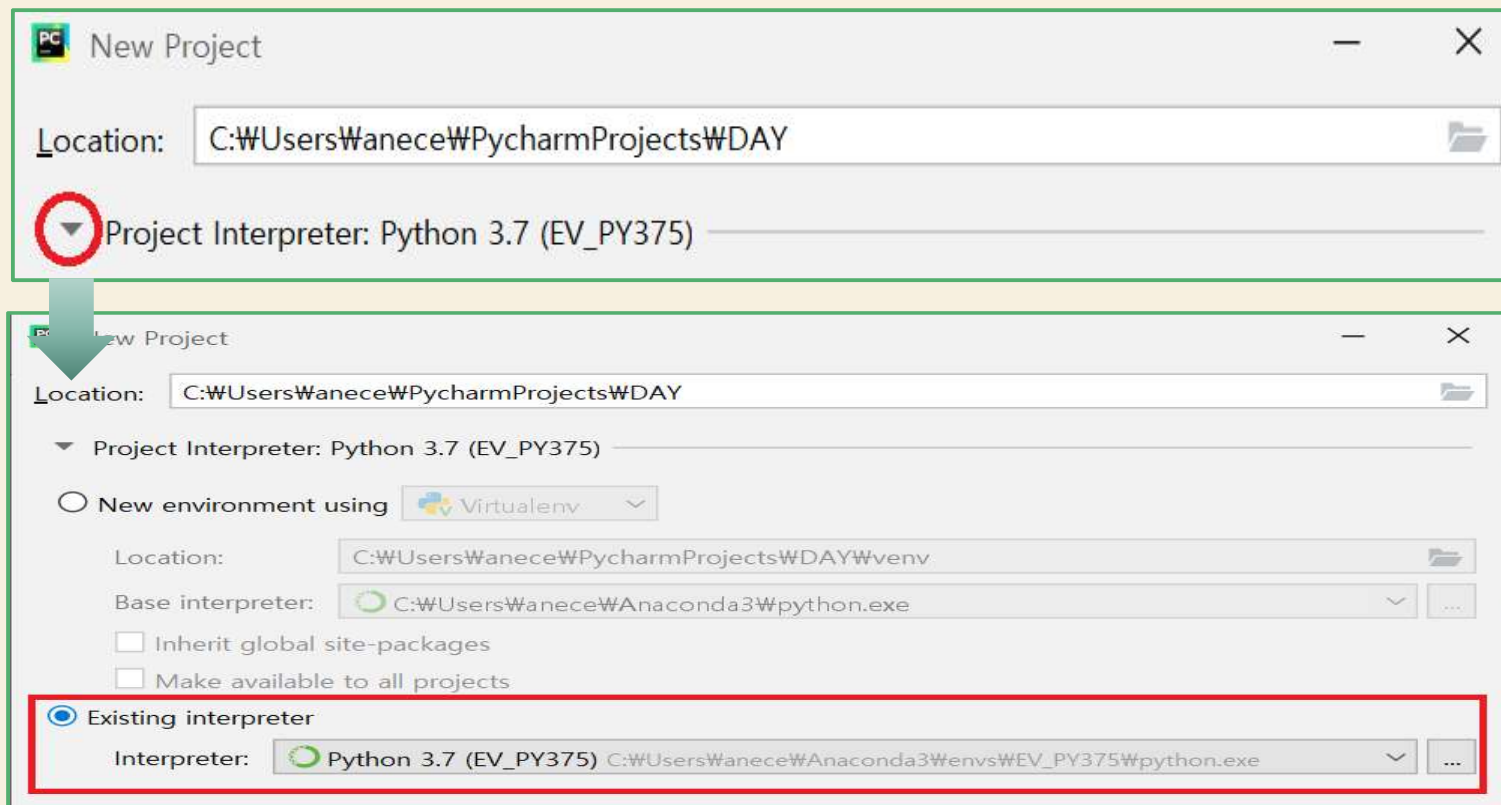
### ➤ Python 인터프리터 설정



# 개발환경구축

## ◆ 아나콘다(Anaconda) & PyCharm 연동

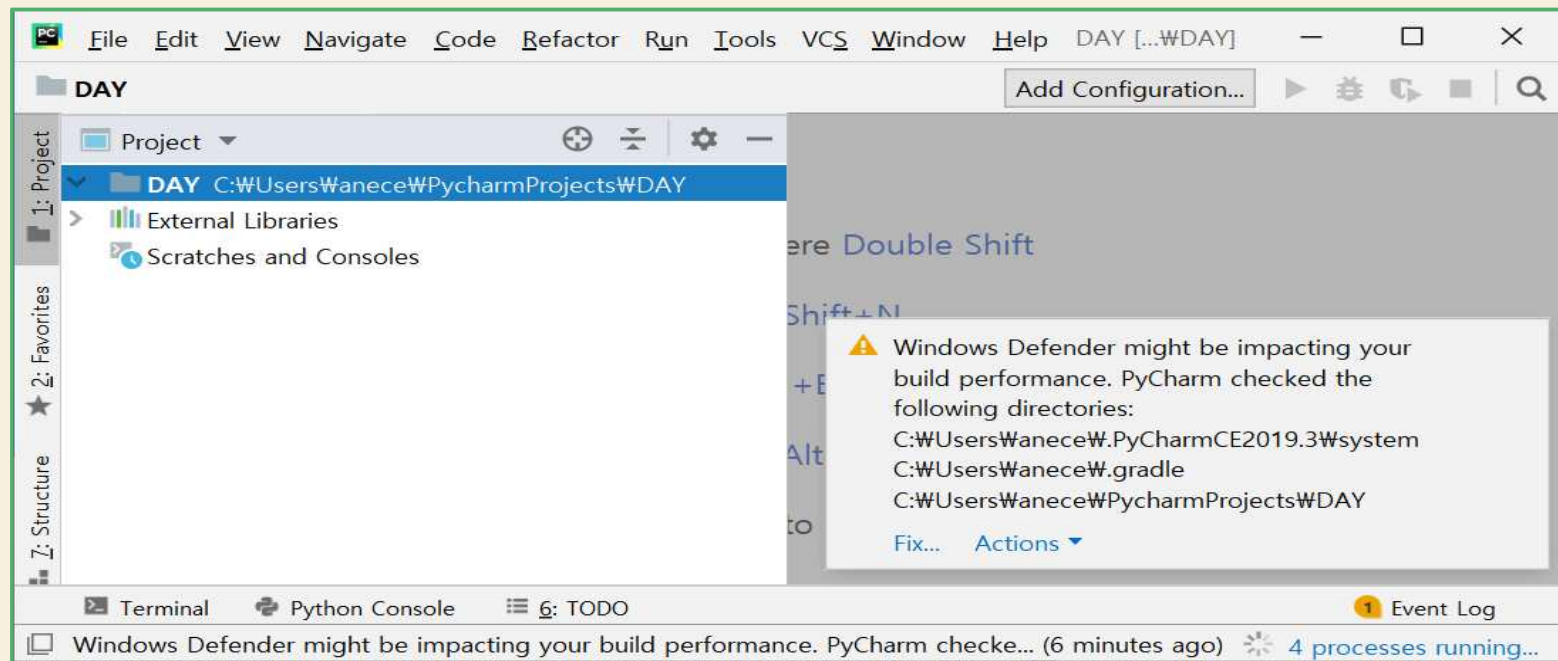
### ➤ 프로젝트 생성



# 개발환경구축

## ◆ 아나콘다(Anaconda) & PyCharm 연동

### ➤ 프로젝트 생성



Windows Defender 혹은 그외 Anti Virus 백신에서 프로젝트 디렉토리를 제외 처리 해야 성능이 좋아진다 ➔ **Fix...**

# WEB 개발도구

## ◆ Jupyter Notebook

- 프로젝트 주피터의 제품 중 하나
  - "오픈 소스 소프트웨어, 개방형 표준, 그리고 여러 개의 프로그래밍 언어에 걸쳐 인터랙티브 컴퓨팅을 위한 서비스 개발"을 위해 설립된 비영리 단체
  - 2014년에 IPython으로부터 파생
- 여러 개의 언어를 통한 실행 환경을 지원
- 인터랙티브 컴퓨팅 제품인 주피터 노트북, 주피터허브, 주피터랩 개발 및 지원



# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ 실행

방법1) 시작 > Anaconda3 (64-bit) > Jupyter Notebook

방법2) C:\Users\<사용자계정>\Anaconda3\python.exe -m notebook

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ 실행

```
Jupyter Notebook (Anaconda3)
[W 2021-08-11 05:22:49.458 LabApp] 'notebook_dir' has moved from NotebookApp to ServerApp. This config
to ServerApp. Be sure to update your config before our next release.
[I 2021-08-11 05:22:49.471 LabApp] JupyterLab extension loaded from C:\ProgramData\Anaconda3\lib\site-
r\lab
[I 2021-08-11 05:22:49.472 LabApp] JupyterLab application directory is C:\ProgramData\Anaconda3\share\l
[I 05:22:49.480 NotebookApp] Serving notebooks from local directory: C:\Users\anece
[I 05:22:49.480 NotebookApp] Jupyter Notebook 6.3.0 is running at:
[I 05:22:49.480 NotebookApp] http://localhost:8888/?token=dbc7c9299660ddb75ce072ca190d6b993b0762a45d8e927a
[I 05:22:49.481 NotebookApp] or http://127.0.0.1:8888/?token=dbc7c9299660ddb75ce072ca190d6b993b0762a45d8e927a
[I 05:22:49.481 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to sk
[C 05:22:49.632 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/anece/AppData/Roaming/jupyter/runtime/nbserver-27228-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=dbc7c9299660ddb75ce072ca190d6b993b0762a45d8e927a
or http://127.0.0.1:8888/?token=dbc7c9299660ddb75ce072ca190d6b993b0762a45d8e927a
```

# WEB 개발도구

## ◆ Jupyter Notebook

➤ 실행 → 브라우저 연동

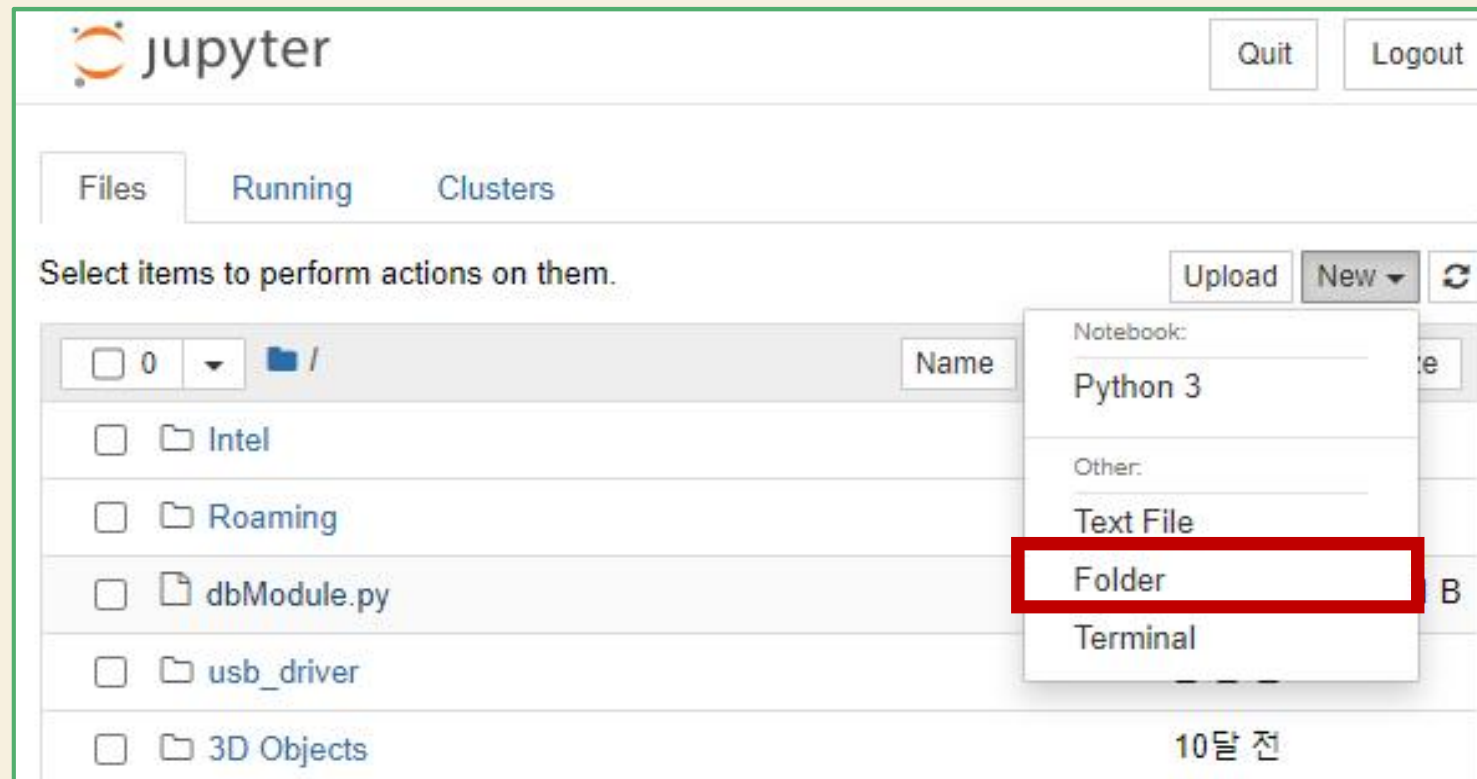




# WEB 개발도구

## ◆ Jupyter Notebook

➤ 작업 폴더 생성

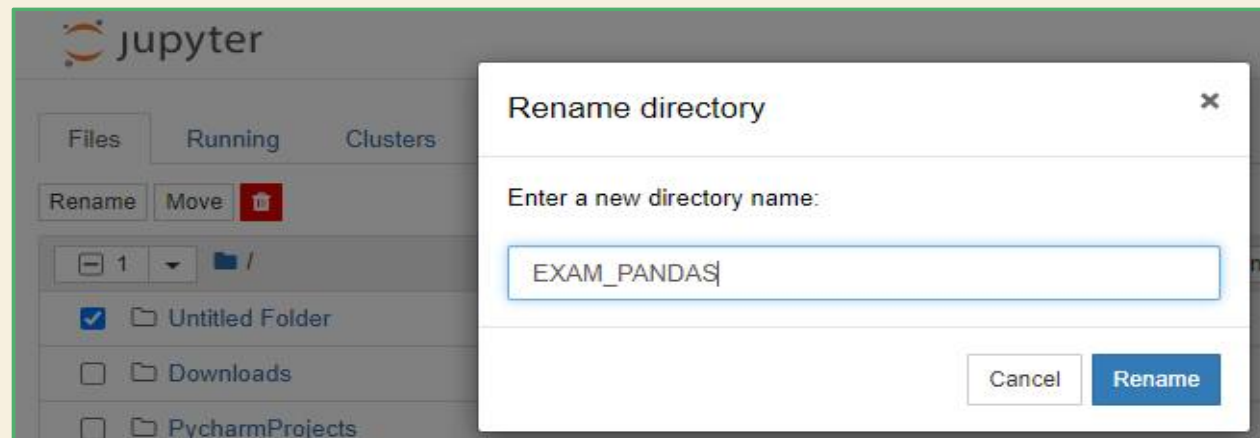




# WEB 개발도구

## ◆ Jupyter Notebook

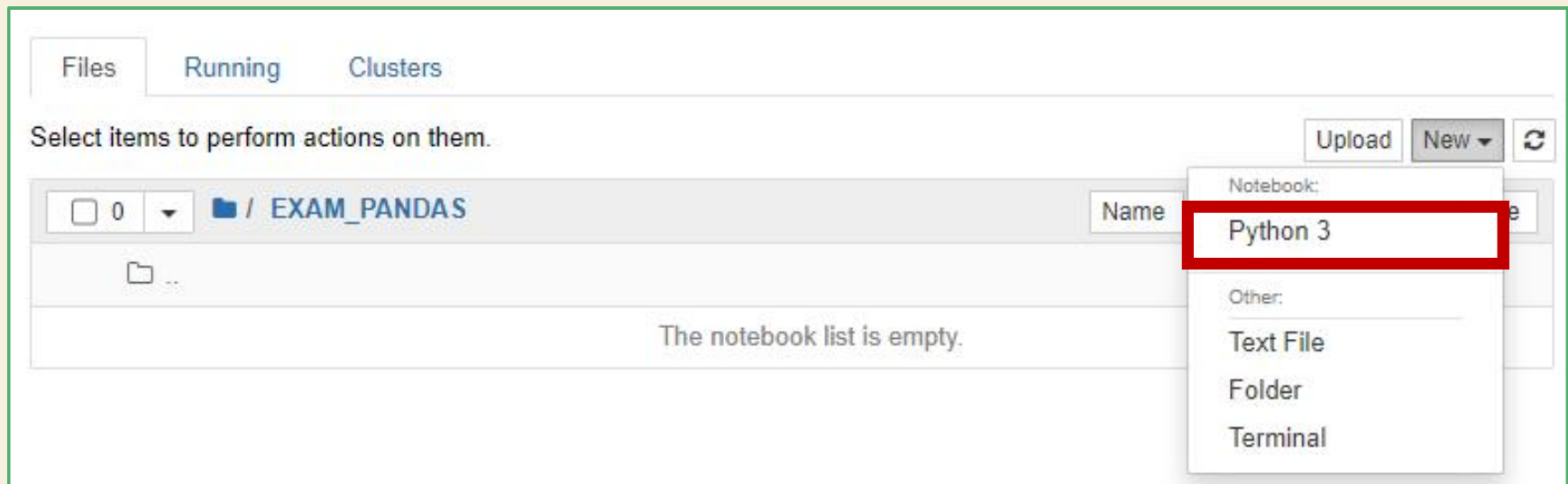
### ➤ 작업 폴더 생성



# WEB 개발도구

## ◆ Jupyter Notebook

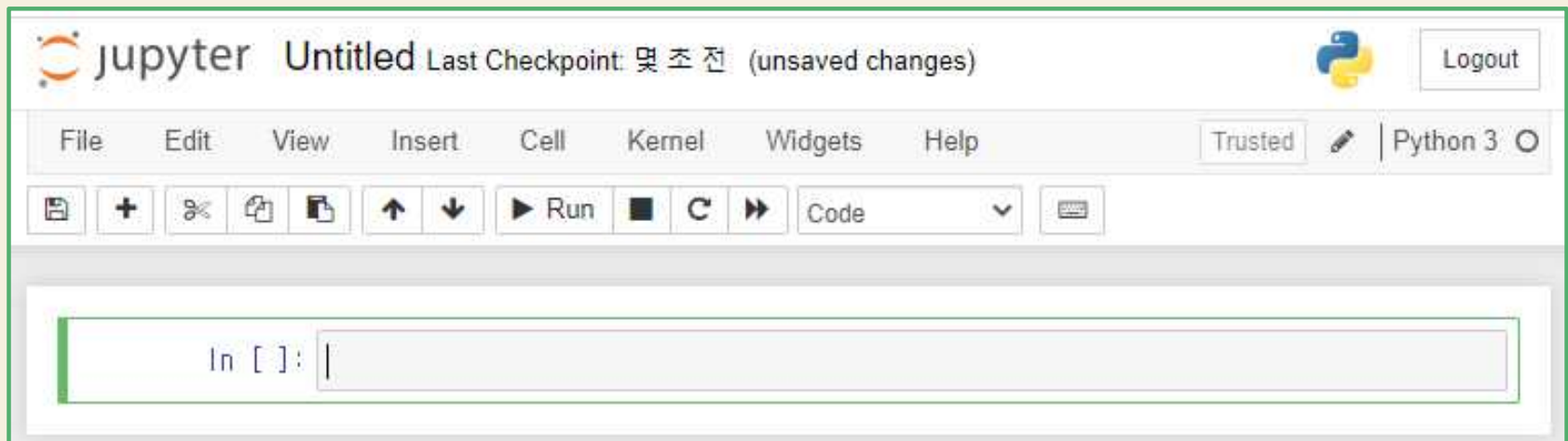
### ➤ 파일 생성



# WEB 개발도구

## ◆ Jupyter Notebook

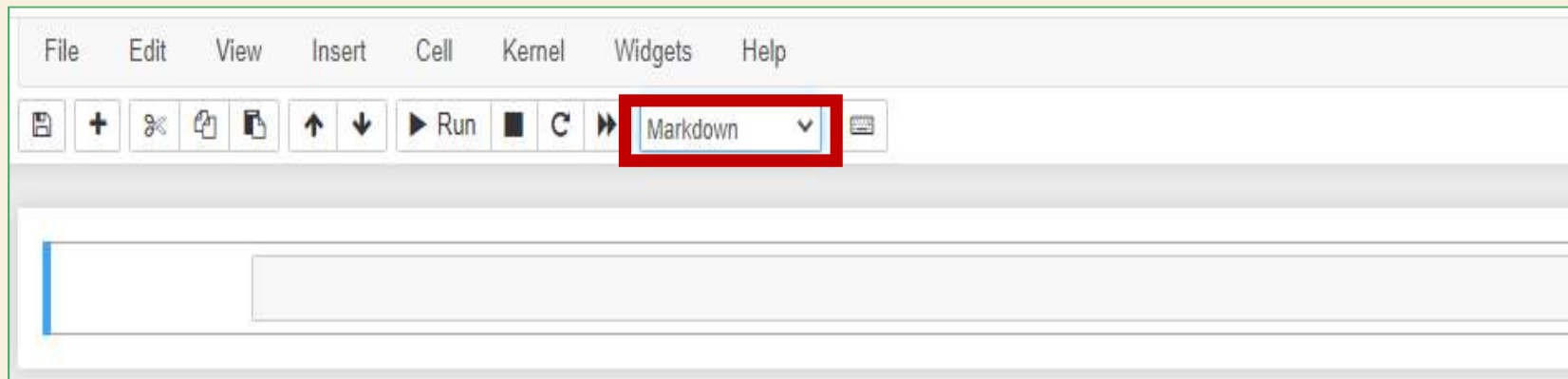
### ➤ 파일 화면



# WEB 개발도구

## ◆ Jupyter Notebook

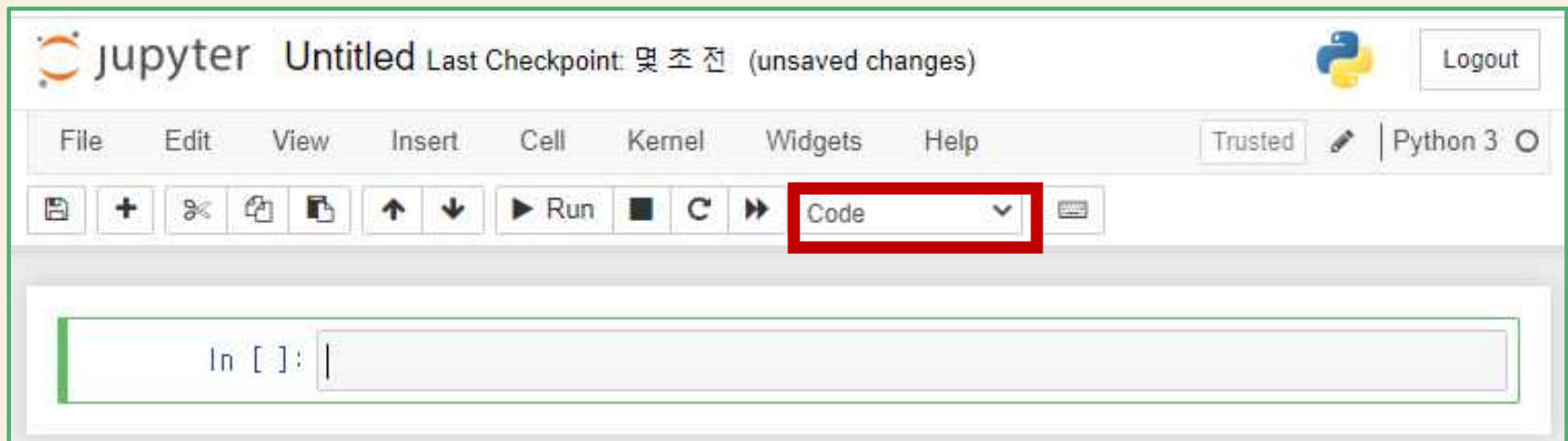
➤ 파일 - 설명 작성



# WEB 개발도구

## ◆ Jupyter Notebook

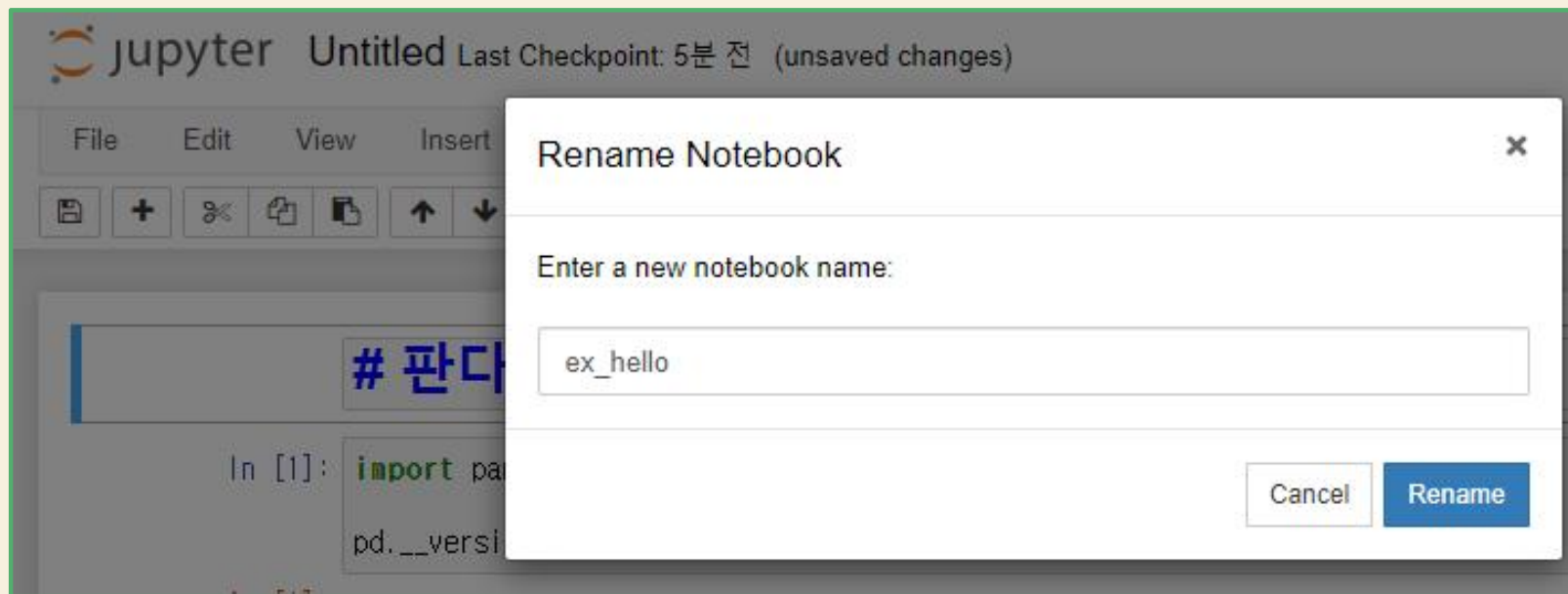
### ➤ 파일 - 코드 작성



# WEB 개발도구

## ◆ Jupyter Notebook

➤ 파일 - 파일명 변경

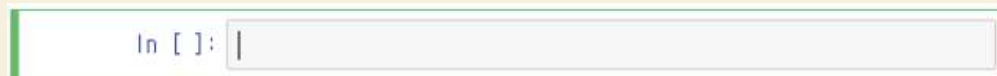


# WEB 개발도구

## ◆ Jupyter Notebook

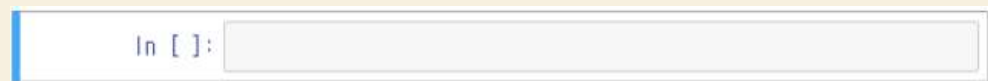
### ➤ 동작 모드

- Edit 모드



- 셀을 클릭하거나 키보드에서 **Enter**키 입력
- 셀 테두리 **녹색**으로 전환
- **셀 입력 가능**

- Command 모드



- 셀 외부의 아무 곳이나 클릭하거나 키보드에서 **Esc**키 입력
- 셀 테두리 **파랑색**으로 전환
- 셀에 내용 입력 불가 → **노트북 편집** 가능

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ 셀(Cell) 다루기

- 셀 추가

- 현재 셀 위에 추가 : command 모드 + a
- 현재 셀 아래에 추가 : command 모드 + b

- 셀 삭제

- 현재 셀 삭제 : command 모드 + dd
- 셀 제거 취소 : command 모드 + z



# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ 셀(Cell) 다루기

- 셀 복사

- 현재 셀 복사 : command 모드 + c

- 셀 붙여넣기

- 현재 셀 위에 붙여 넣기: command 모드 + Shift+v
- 현재 셀 아래에 붙여 넣기 : command 모드 + v

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ 셀(Cell) 다루기

- 셀 실행

- 현재 셀 실행 : command 모드 + Ctrl + Enter
- 현재 셀 실행 + 다음 셀 이동 : command 모드 + Shift + Enter

- 입력모드 변경

- 현재 셀 Markdown으로 변경 : command 모드 + m
- 현재 셀 Code로 변경 : command 모드 + y

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ 패키지 관리

- 확인

```
In [ ]: !pip list
```

- 설치

```
In [ ]: !pip install package_name
```

- 업그레이드

```
In [ ]: !pip install package_name --upgrade
```

- 정보 보기

```
In [ ]: !pip show pandas
```

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

- 텍스트 기반의 **마크업언어**
- 2004년 존그루버가 만들었고 쉽게 쓰고 읽을 수 있으며 HTML 변환 가능
- **특수기호와 문자**를 이용한 매우 **간단한 구조의 문법** 사용
- 웹에서도 보다 빠르게 콘텐츠를 작성, 직관적으로 인식

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

#### 헤더 Headers

- # H1
- ## H2
- ### H3
- #### H4
- ##### H5
- ##### H6

```
# This is a H1
## This is a H2
### This is a H3
#### This is a H4
##### This is a H5
##### This is a H6
```

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

#### 블럭쿼터 BlockQuote

- 인용 블록 의미 >
- 안에 다른 Markdown 요소 포함 가능

```
> This is a first blockquote.  
>     > This is a second blockquote.  
>     >     > This is a third blockquote.
```

This is a first blockquote.

This is a second blockquote.

This is a third blockquote.

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

#### 목록 List

- 순서있는 목록(번호)

- 숫자. 항목명

```
1. 첫 번째  
2. 두 번째  
3. 세 번째
```

```
1. 첫 번째  
2. 두 번째  
3. 세 번째
```

- 순서없는 목록

- \*, +, - 항목명

```
* 빨강  
* 녹색  
* 파랑
```

```
+ 빨강  
+ 녹색  
+ 파랑
```

```
- 빨강  
- 녹색  
- 파랑
```

```
• 빨강  
  ○ 녹색  
    ■ 파랑
```

```
• 빨강  
  ○ 녹색  
    ■ 파랑
```

```
• 빨강  
  ○ 녹색  
    ■ 파랑
```

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

#### 목록 List

- 순서있는 & 순서없는 혼합

- \* 1단계
  - 2단계
    - + 3단계
      - + 4단계

- 1단계
  - 2단계
    - 3단계
      - 4단계



# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

들여쓰기 Intention

- 4개 공백 또는 하나의 탭 + 한 줄 띄어쓰기

This is a normal paragraph:

공백

This is a code block.

공백

end code block.

This is a normal paragraph:

This is a code block.

end code block.

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

코드블럭

- 방법 1) `<pre><code>{code}</code></pre>`

```
<pre>
<code>
public class BootSpringBootApplication {
    public static void main(String[] args) {
        System.out.println("Hello, Honeymon");
    }
}
</code>
</pre>
```

```
public class BootSpringBootApplication {
    public static void main(String[] args) {
        System.out.println("Hello, Honeymon");
    }
}
```

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

코드블럭

- 방법 2) ""코 드 ""

```
...  
public class BootSpringBootApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello, Honeymon");  
    }  
}  
...
```

```
public class BootSpringBootApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello, Honeymon");  
    }  
}
```

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

코드블럭

- 방법 2) ""코 드 ""

문법강조(Syntax highlighting)

```
```java
```

```
public class BootSpringBootApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello, Honeymon");  
    }  
}  
...  
`
```

```
public class BootSpringBootApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello, Honeymon");  
    }  
}
```

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

줄바꿈 / 개행

- 띄어쓰기 2번 입력

```
good Luck Happy  
2022 2022 2022  
Merry Christmas  
HaHaHaHa
```

```
good Luck Happy 2022 2022 2022  
Merry Christmas  
HaHaHaHa
```

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

문장 내 강조

- 볼드(진하게) → `_진하게_`
- 이탤릭 → `_이탤릭 기울여서_`
- 취소선 → `~~취소~~`
- 밑줄 → `<u>밑줄</u>`

**볼드(진하게)**  
*이탤릭체(기울여서)*  
~~취소선~~  
밑줄

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

수식

- \$ 수식 \$ : 왼쪽 정렬
- \$\$ 수식 \$\$ : 가운데 정렬

```
$y=ax+b$  
$$y=ax+b$$
```

$$y = ax + b$$

$$y = ax + b$$

# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

수평선

- 미리보기로 출력할 때 페이지 나누기 용도로 많이 사용





# WEB 개발도구

## ◆ Jupyter Notebook

### ➤ Markdown

#### 매직명령어

- 터미널 명령어 사용
- 형식 : %명령어
- %pwd
- %ls
- %mkdir %rmdir
- %load 파일명 => 파일 내용 코드셀로 출력
- %%writefile 파일명 => 파일 저장

```
In [1]: %lsmagic
Out[1]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave
%copy %ddir %debug %dhist %dirs %doctest_mode %echo %ed %edit
%load_ext %loadpy %logout %logon %logstart %logstate %logstop
%page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %pip
%pycat %pylab %qtconsole %quickref %recall %rehashx %reload_ext
%sc %set_env %store %sx %system %tb %time %timeit %unalias

Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%cnd %%debug %%file %%html
python %%python2 %%python3 %%ruby %%script %%sh %%svg %%sx %

Automagic is ON, % prefix IS NOT needed for line magics.
```

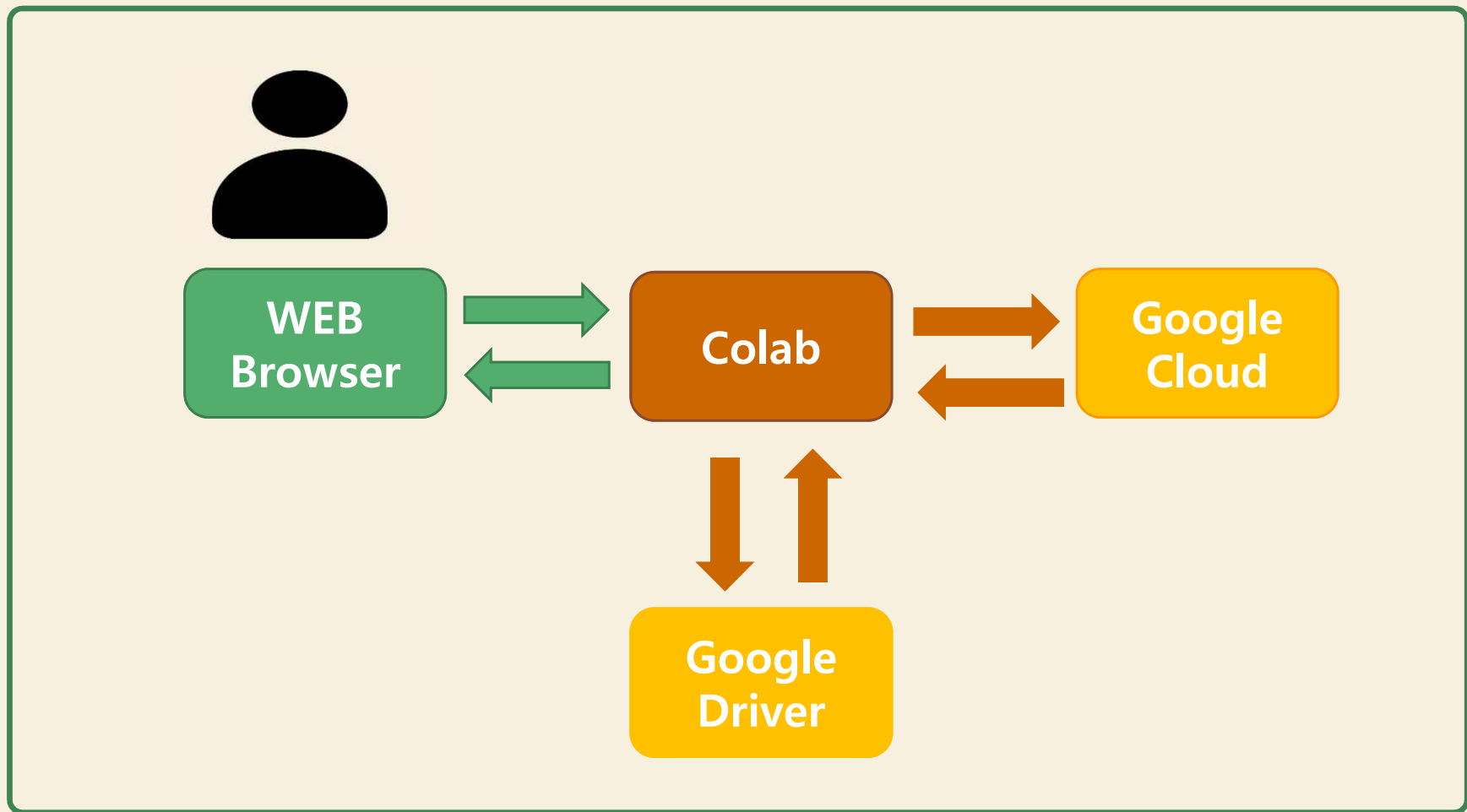
# WEB 개발도구

## ◆ 구글 코랩(CoLab)

- Google Laboratory 서비스 줄임말
- 클라우드 기반으로 주피터 노트북 개발환경
- 교육과 과학 연구를 목적으로 개발된 도구
- 2017년 무료 공개
- 구글 드라이브, 도커, 리눅스, 구글 클라우드 기술로 이루어짐
- 저 사양 컴퓨터이거나 GPU가 없는 컴퓨터에 추천
- 데이터 분석 및 AI를 위한 패키지 기본 설치되어 있음
- 단, 원하는 버전의 환경 사용은 불가

# WEB 개발도구

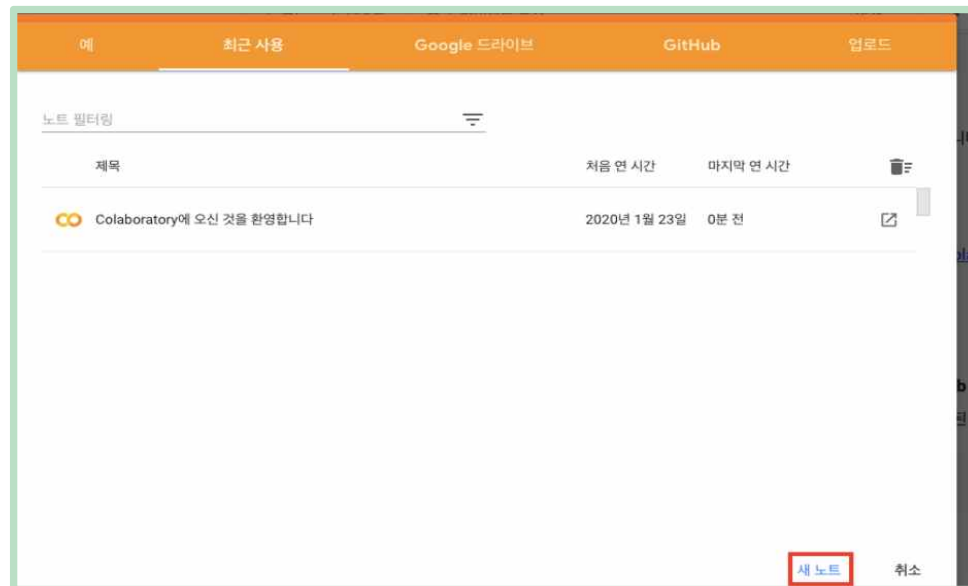
## ◆ 구글 코랩(CoLab)



# WEB 개발도구

## ◆ 구글 코랩(CoLab)

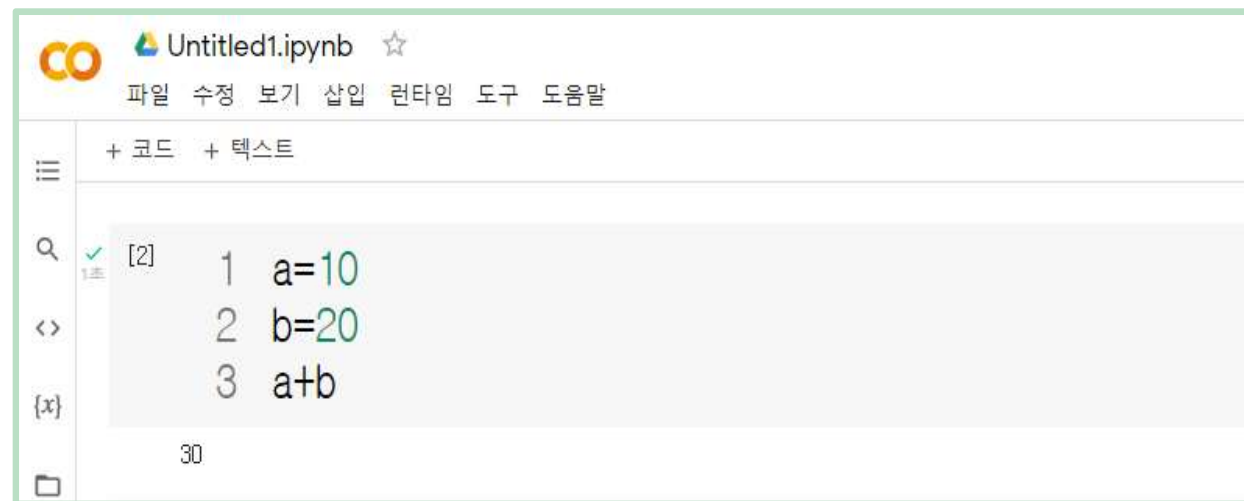
- Google 회원가입 & 로그인
- 새노트 클릭



# WEB 개발도구

## ◆ 구글 코랩(CoLab)

- 입력 후 실행 → **Shift+Enter**



The screenshot shows the Google Colab web interface. At the top, there's a header with the Colab logo, the file name 'Untitled1.ipynb', and a star icon. Below this is a menu bar with options: '파일' (File), '수정' (Edit), '보기' (View), '삽입' (Insert), '런타임' (Runtime), '도구' (Tools), and '도움말' (Help). The main area has a sidebar on the left with icons for file explorer, search, code editor, and output. The code editor shows a code cell with the following code:

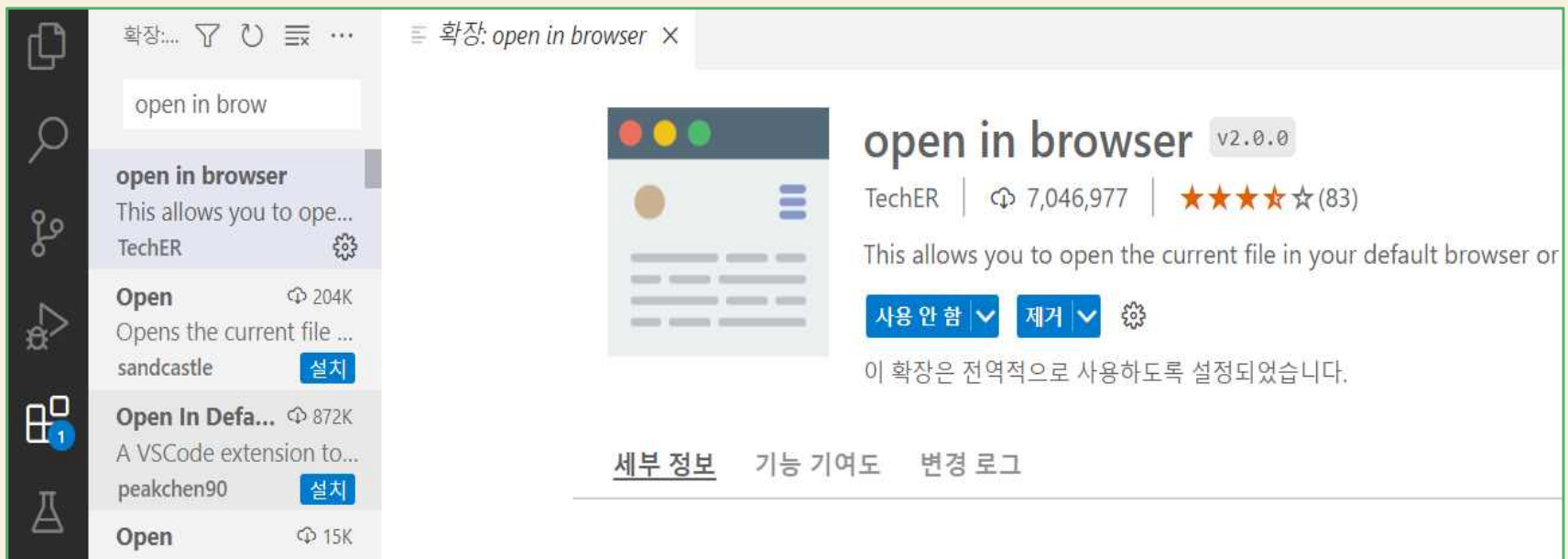
```
[2] 1 a=10
      2 b=20
      3 a+b
```

The output of the code cell is displayed below the code, showing the result of the addition: 30.

# 개발환경구축

## ◆ VSCODE

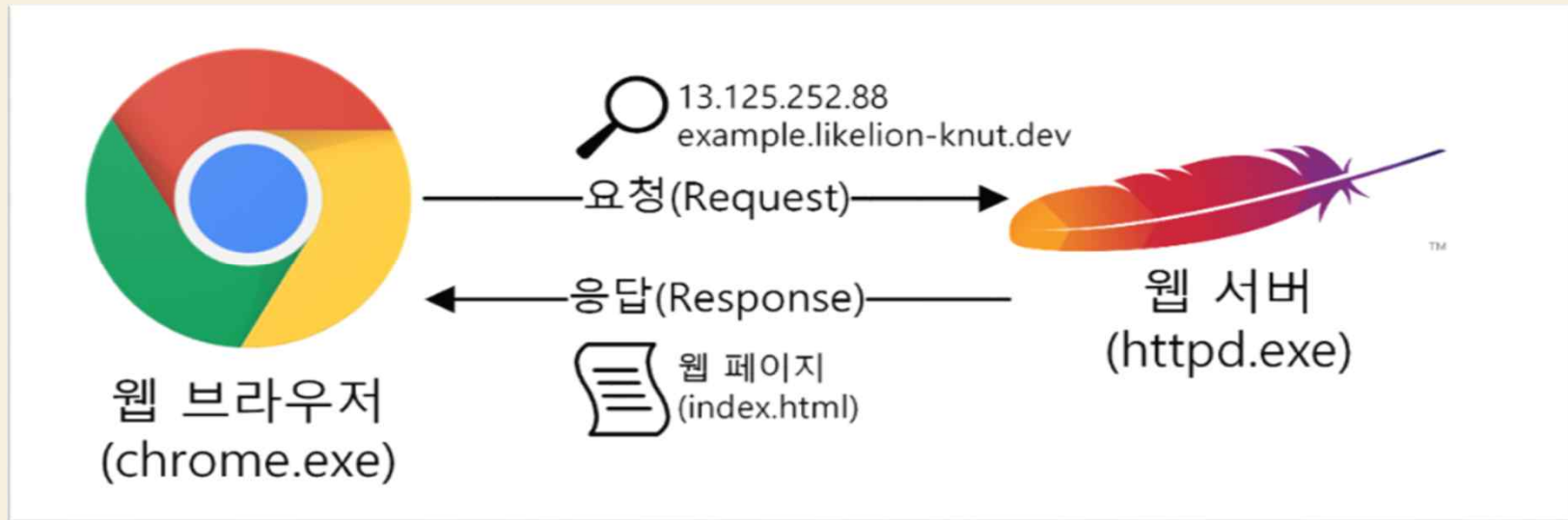
### ➤ 확장 프로그램 설치



# CH02 – WEB FRONTEND

# WEB FRONTEND

## ◆ WEB 동작원리



- 웹 브라우저와 웹 서버는 HTTP 프로토콜 이용 통신
- 웹 브라우저에서 특정 UR을 웹서버에 요청
- 웹 서버는 해당 어떤 디렉토리에서 특정 HTML 파일 검색
- 다시 웹 브라우저(Web Browser)에게 제공
- 웹 브라우저는 제공받은 HTML파일을 사용자의 화면에 출력



# WEB FRONTEND

## ◆ HTML5

- "Hyper Text Markup Language" 5의 약자
- HTML 4.01, XHTML 1.1 등 대체하는 차세대 웹표준(2014/10/28)

이전 HTML	HTML 5
<ul style="list-style-type: none"><li>• 비디오 , 오디오 등의 기능 처리하기 위해선 플러그인이라는 별도의 프로그램 필요</li><li>• 문법을 잘 지켜서 코딩</li><li>• 텍스트와 하이퍼링크만 표시</li></ul>	<ul style="list-style-type: none"><li>• 비디오 , 오디오 등 기능 처리 가능</li><li>• 문법적으로 매우 유연하게 대처<ul style="list-style-type: none"><li>✓ Tag 이름에 대문자 사용</li><li>✓ empty Tag의 종료 Tag 생략 가능</li></ul></li></ul>

# WEB FRONTEND

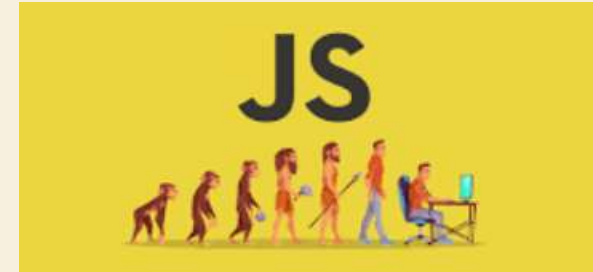
## ◆ CSS3



- "Cascading Style Sheets"의 약자
- 웹의 전반적인 스타일을 미리 저장해 둔 스타일 시트
- 기존 HTML의 경우 다양하게 설계하고 변경하는데 많은 제약
- 보완하기 위해 만들어진 것이 스타일 시트
- 스타일 시트 표준안 CSS
- 그림자 효과, 그라데이션, 변형 등 다양한 애니메이션 기능 추가

# WEB FRONTEND

## ◆ JavaScript



- 초창기 자바스크립트는 웹 페이지의 보조적인 기능 수행 한정적인 용도
- 1999년, 자바스크립트 이용해서 비동기적(Asynchronous)으로 서버와 브라우저가 데이터를 교환할 수 있는 통신 기능인 **Ajax** 등장
- Ajax의 등장은 패러다임을 획기적으로 전환 => **필요한 부분만 한정적으로 렌더링**
- 2005년, **Google Maps**는 **Ajax+Javascript** 구현
- 2006년, jQuery의 등장, 자바스크립트보다 배우기 쉽고 직관적인 jQuery를 더 선호
- 2009년, 브라우저 이외의 환경에서 동작시킬 수 **Node.js** 등장으로 자바스크립트는 웹 브라우저를 벗어나 서버 사이드 애플리케이션 개발

# WEB FRONTEND

## ◆ JavaScript

- HTML, CSS와 함께 웹을 구성하는 요소 중 하나로 **웹 브라우저에서 동작하는 유일한 프로그래밍 언어**
- **기본 문법은 C, Java**와 유사하고 Self에서는 프로토타입 기반 상속을, Scheme에서는 일급 함수의 개념을 차용
- 개발자가 별도의 컴파일 작업을 수행하지 않는 **인터프리터 언어(Interpreter language)**
- **자바스크립트 엔진** => 인터프리터 + 컴파일 장점 결합 → 느린 속도 해결
- 명령형(imperative), 함수형(functional), 프로토타입 기반(prototype-based) 객체지향 프로그래밍 지원하는 멀티 패러다임 프로그래밍 언어
- 다른 객체지향 언어들과의 차이점에 대한 논쟁들이 있음

# WEB FRONTEND

## ◆ HTML 구조

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE> New Document </TITLE>
    <META NAME="Generator" CONTENT="EditPlus">
    <META NAME="Author" CONTENT="">
    <META NAME="Keywords" CONTENT="">
    <META NAME="Description" CONTENT="">
  </HEAD>

  <BODY bgcolor='yellow'>
    <h1> HEADER </h1>
    <a href="http://www.naver.com">GO~! NAVER</a>

    <div>
      SPACE 1
    </div>

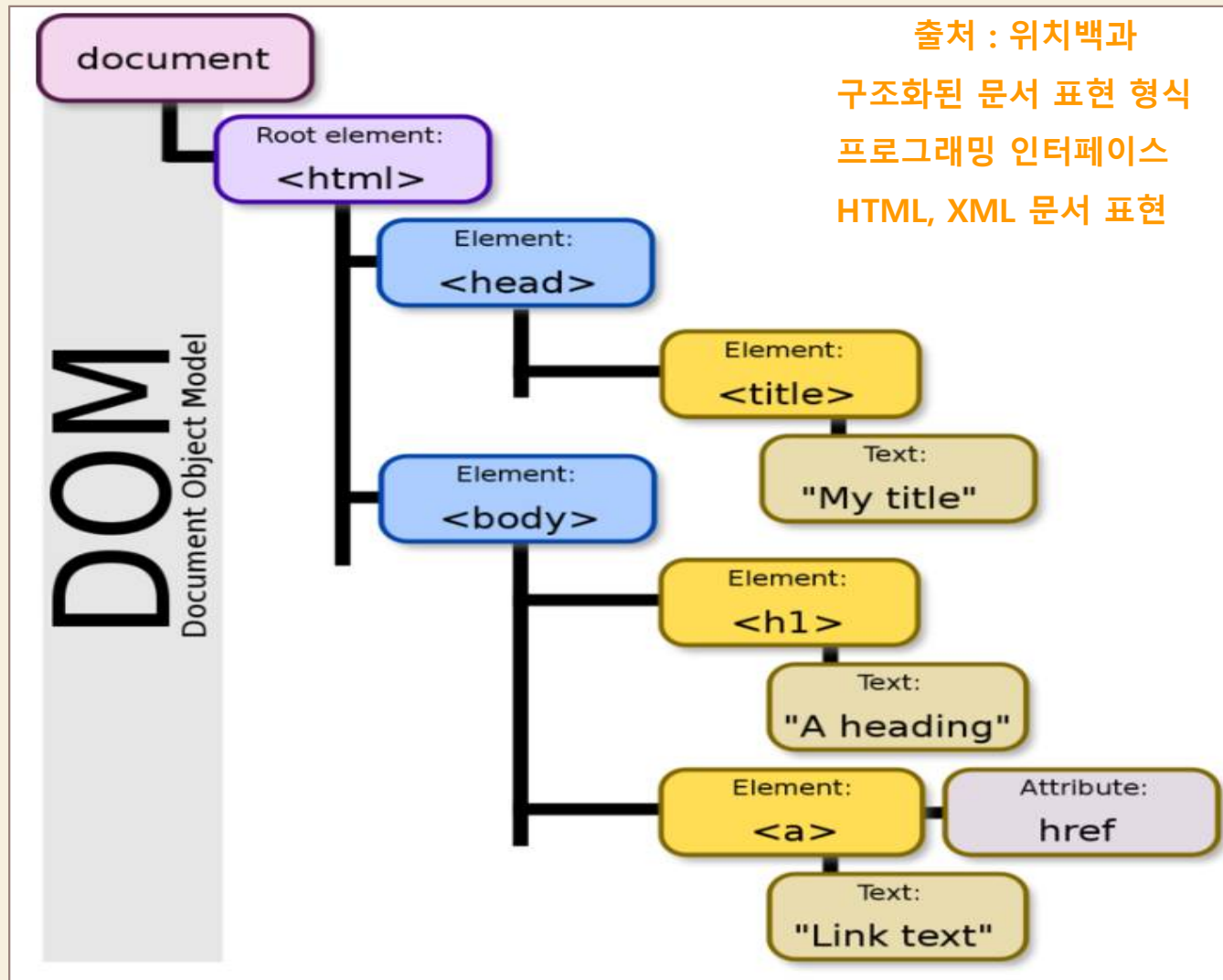
    <div>
      SPACE 2
    </div>
  </BODY>
</HTML>
```

### 태그(Tag)

- HTML 구성 요소
- 사용자에게 보여줄 여러가지 요소
- 형식 : < 태그명 > </태그명>
- 역할에 따라 다양한 속성 존재

# WEB FRONTEND

## ◆ DOM 구조



# WEB FRONTEND

## ◆ HTML

- **<style> </style> 태그**

- ✓ 문서나 문서 일부에 대한 스타일 정보를 포함
- ✓ <head> 안에 위치

```
<head>  
  <style>  
    body {  
      background_color:#FFFF00;  
    }  
  </style>  
</head>
```

# WEB FRONTEND

## ◆ CSS

➤ <https://developer.mozilla.org/ko/docs/Web/CSS>

### ◆ 선택자 (Selector)

HTML파일에서 특정 태그를 지정하기 위해서 사용하는 방법

선택자		의미
전체 선택자	*	웹 페이지 안의 전체 요소 대상
타입 선택자	태그명	모든 요소들 대상 의미
속성 선택자	태그[속성명], 태그[속성명=값]	속성에 대항하는 대상 의미
아이디 선택자	#아이디명	id속성을 가지는 대상, 페이지 내에 1개만 존재
클래스 선택자	.클래스명	class속성을 가지는 대상, 페이지 내에 여러 개 존재
하위 선택자	조상 태그 자손 태그	태그 아래 자손 태그들 의미, 여러 개 존재
자식 선택자	부모 태그 > 자식 태그	부모 아래 바로 존재하는 태그



# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 적용

1. 내부 자바스크립트 코드 적용, <head>, <body> 모두 위치 가능

```
<script>
```

```
    document.getElementById("text").innerHTML = "환영합니다!";
```

```
</script>
```

2. 외부 자바스크립트 파일로 적용

```
<head>
```

```
    <title>JavaScript Apply</title>
```

```
    <script src="/examples/media/example.js"></script>
```

```
</head>
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 문장

```
var fruit ='banana' ;
```

- 주석

```
// 한 줄 주석
```

```
/*  
여러줄 주석  
*/
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 변수 선언

① 변수명 = 초기값; // 전역변수  
school = '대구';

② var 선언자 ➔ var 변수명; var 변수명 = 초기값; // 전역, 지역변수  
var name ;  
name='마징가'  
var name = "김철수"; // 중복 선언 가능

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 변수 선언

③ let 선언자 => 중복 선언 불가, 전역/지역변수

```
let name;
```

```
name = " 홍길동"
```

```
let pear = '배';
```

④ const 선언자 => 상수 선언

선언과 동시에 초기화, 값 변경 불 가능

```
const NAME = "마징가";
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 데이터 타입

```
var num = 10;                // 숫자
var num = 10.001;

var myName = "홍길동";       // 문자열
var myName = '홍길동';

var save = false;            // 논리형

var str;                     // undefined

var str = null;              // object 타입, 값이 null
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => window.alert() 메소드

```
<script>  
  function alertDialogBox() {  
    alert("확인을 누를 때까지 다른 작업을 할 수 없어요!");  
  }  
</script>
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => document.write() 메소드

웹 페이지가 로딩될 때 실행, 웹 페이지에 가장 먼저 데이터 출력  
대부분 테스트나 디버깅을 위해 사용

```
<script>  
    document.write("Loading !");  
</script>
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => console.log() 메소드

웹 브라우저의 콘솔을 통해 데이터를 출력

대부분 주요 웹 브라우저는 F12 누>> 콘솔 메뉴 클릭 >> 콘솔 화면 사용

```
<script>  
    console.log("Loading !");  
</script>
```



# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => HTML DOM 요소 이용 innerHTML 프로퍼티

### ◆ DOM 요소의 선택

1. HTML 태그 이름(tag name) 이용한 선택
2. 아이디(id)를 이용한 선택
3. 클래스(class)를 이용한 선택
4. name 속성(attribute)을 이용한 선택
5. CSS 선택자(selector)를 이용한 선택
6. HTML 객체 집합(object collection)을 이용한 선택

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => HTML DOM 요소 이용 innerHTML 프로퍼티

### ◆ DOM 요소의 선택

1. HTML 태그 이름(tag name) 이용한 선택

```
var selectedItem = document.getElementsByTagName("li");  
for (var i = 0; i < selectedItem.length; i++) {  
    selectedItem.item(i).style.color = "red";  
}
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => HTML DOM 요소 이용 innerHTML 프로퍼티

### ◆ DOM 요소의 선택

2. 아이디(id)를 이용한 선택

```
// 아이디가 "even"인 요소를 선택함.  
var selectedItem = document.getElementById("even");  
  
selectedItem.style.color = "red";
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => HTML DOM 요소 이용 innerHTML 프로퍼티

### ◆ DOM 요소의 선택

#### 3. 클래스(class)를 이용한 선택

```
// 클래스가 "odd"인 모든 요소를 선택함.  
var selectedItem = document.getElementsByClassName("odd");  
  
for (var i = 0; i < selectedItem.length; i++) {  
    selectedItem.item(i).style.color = "red";  
}
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => HTML DOM 요소 이용 innerHTML 프로퍼티

### ◆ DOM 요소의 선택

4. name 속성(attribute)을 이용한 선택

```
// name 속성값이 "first"인 모든 요소를 선택함.  
var selectedItem = document.getElementsByName( "first" );  
  
for (var i = 0; i < selectedItem.length; i++) {  
    selectedItem.item(i).style.color = "red";  
}
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => HTML DOM 요소 이용 innerHTML 프로퍼티

### ◆ DOM 요소의 선택

5. CSS 선택자(selector)를 이용한 선택

```
// 클래스가 "odd"인 요소 중에서 <li> 요소만을 선택함.  
var selectedItem = document.querySelectorAll("li.odd");  
  
for (var i = 0; i < selectedItem.length; i++) {  
    selectedItem.item(i).style.color = "red";  
}
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => HTML DOM 요소 이용 innerHTML 프로퍼티

### ◆ DOM 요소의 선택

6. HTML 객체 집합(object collection)을 이용한 선택

```
// <title> 요소를 선택함.  
var title = document.title;  
document.write(title);
```

# WEB FRONTEND

## ◆ JavaScript

❖ [https://developer.mozilla.org/ko/docs/Learn/JavaScript/First\\_steps/A\\_first\\_splash](https://developer.mozilla.org/ko/docs/Learn/JavaScript/First_steps/A_first_splash)

- 출력 => HTML DOM 요소 이용한 innerHTML 프로퍼티

```
<script>  
  function alertDialogBox() {  
    alert("확인을 누를 때까지 다른 작업을 할 수 없어요!");  
  }  
</script>
```