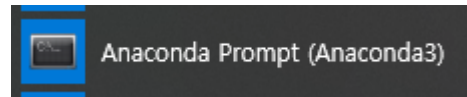


깃허브 개요와 원격 저장소 복사(clone) 활용

동양미래대학교 강환수 교수

복습: 명령 프롬프트 사용

- Path 설정이 매우 중요
 - path
 - where pip
 - where python
- 아나콘다의 path 설정을 안했다면
 - 아나콘다 설치 메뉴, 아나콘다 프롬프트를 사용이 가능 편리



Git과 github

• Git

- 2005년 리누스 토발즈가 개발한 분산 소스 버전 관리 시스템
 - 또는 형상 관리 도구(Configuration Management Tool)
 - 여러 명의 개발자(분산)가 특정 프로젝트를 자신의 컴퓨터로 협업하여 개발하면서 버전을 관리할 수 있는 시스템
 - 소스코드를 주고 받을 필요 없이, 같은 파일을 여러 명이 동시에 작업하는 병렬 개발이 가능
- 설치
 - git-scm.com의 git 설치
- GUI 도구
 - GitHub Desktop, SourceTree, GitKraken
- <https://git-scm.com/book/ko/v2>
 - 공식 Site 한글 매뉴얼

• Github

- 버전 관리 시스템을 지원하는 웹호스팅 서비스
 - 깃을 활용한 파일 서비스 시스템
 - 초보자에겐 파일 서버로 활용
 - 자신의 포트폴리오 사이트로 활용
- GitHub, GitLab, BitBucket 등



깃허브

- 개발자들의 홈페이지로 활용
- 검색 내용

GitHub is how people build software

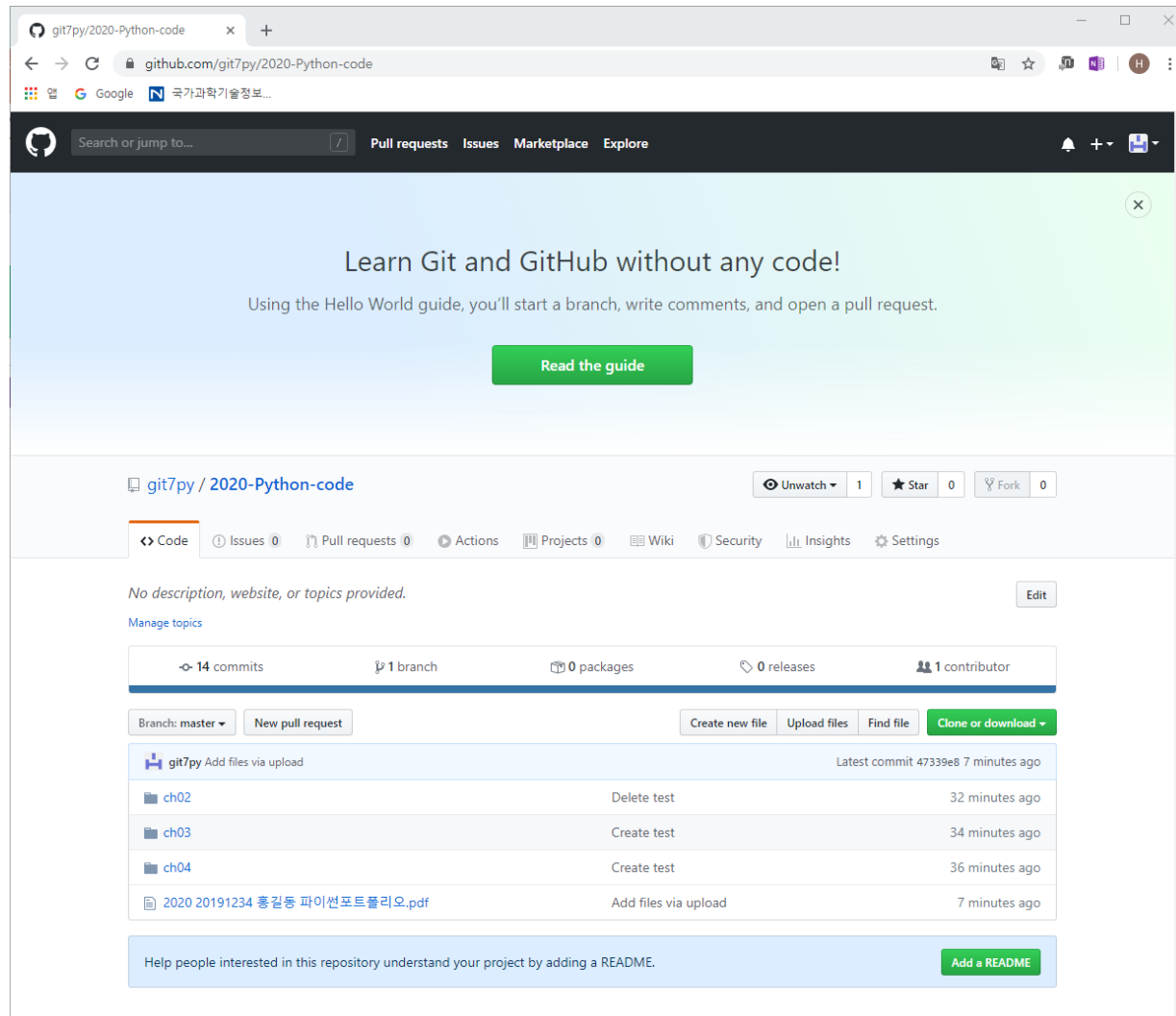
GitHub는 사람들이 소프트웨어를 개발하는 방법입니다

대표적인 무료 Git 저장소. 2008년 공개했다. Git 호스팅 기능 덕분에 GitHub는 자유 소프트웨어의 성지로 떠올랐다. 본사는 미국 샌프란시스코에 있다. 경쟁사 SourceForge는 애드웨어, 해킹, 바이러스 등등의 문제로 많은 개발자들의 신뢰를 잃어 사실상 망했으며, 대부분의 프로젝트가 GitHub 쪽으로 이주하게 됐다. 마스코트는 고양이와 문어를 합친 모습의 Octocat이다.

원래는 공개 프로젝트만 무료였고 비공개 프로젝트는 결제를 해야 했으나, MS에 인수된 이후 2019년 1월 초부터 비공개 저장소를 무료로 제공하기 시작했고 #, 2020년 4월 중순부터 비공개 저장소 공동 작업자수 제한도 풀렸다. #

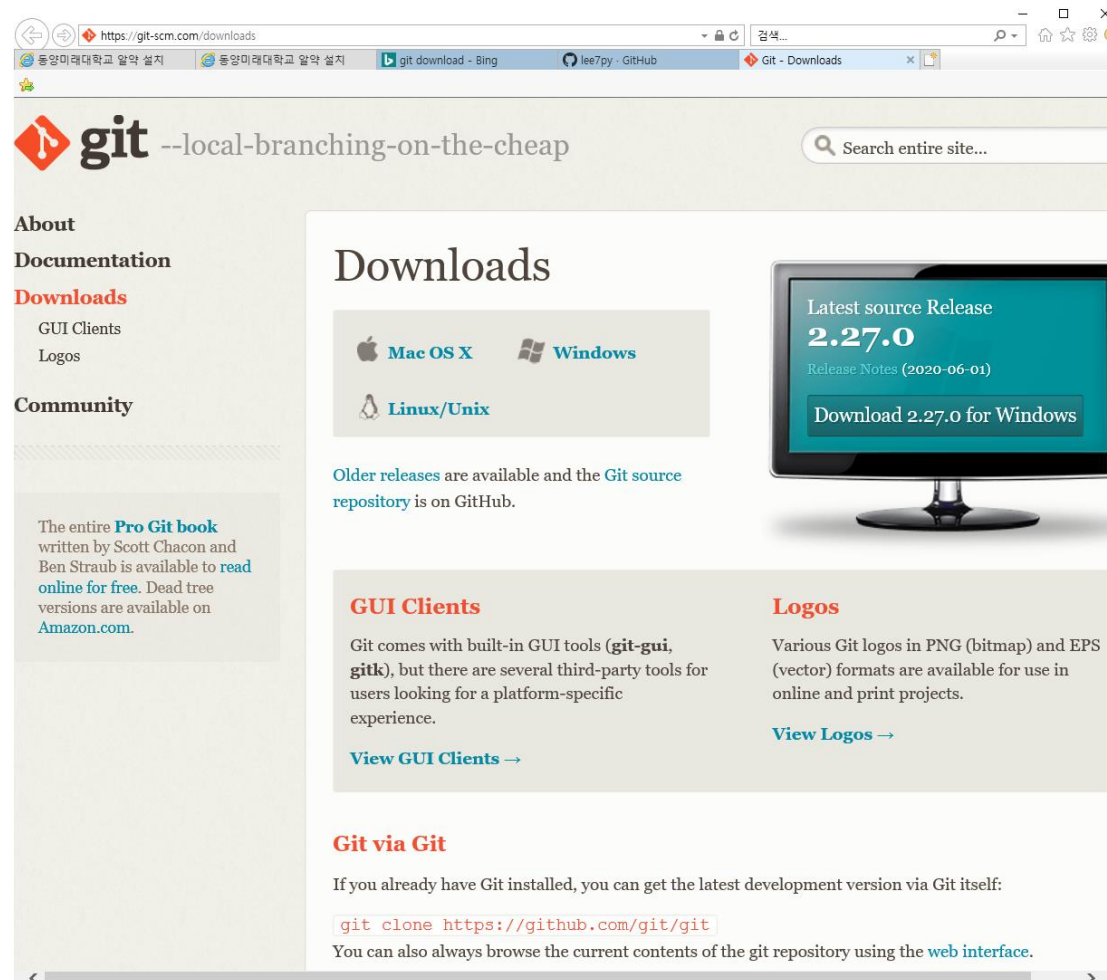
강력한 기능과 안정성을 원한다면 마이크로소프트의  Azure DevOps가 대체재가 될 수 있다. 한 저장소당 최대 5명의 사용자가 무료 이용이 가능하며 공개/비공개 설정도 자유롭다. 그 외에  Bitbucket과 GitLab이라는 후보도 있다. 이 중에서 GitLab이 Bitbucket보다 여러 모로 강력한 기능을 제공하고 제한도 적게 걸지만^[1], Bitbucket에 비하면 서버 운영이 불안정한 편이다. Bitbucket은 Jira와 Confluence로 유명한 Atlassian에서 개발했기 때문에 신뢰도는 확실하다.

깃허브 접속 화면



깃 홈페이지

- <https://git-scm.com/downloads>



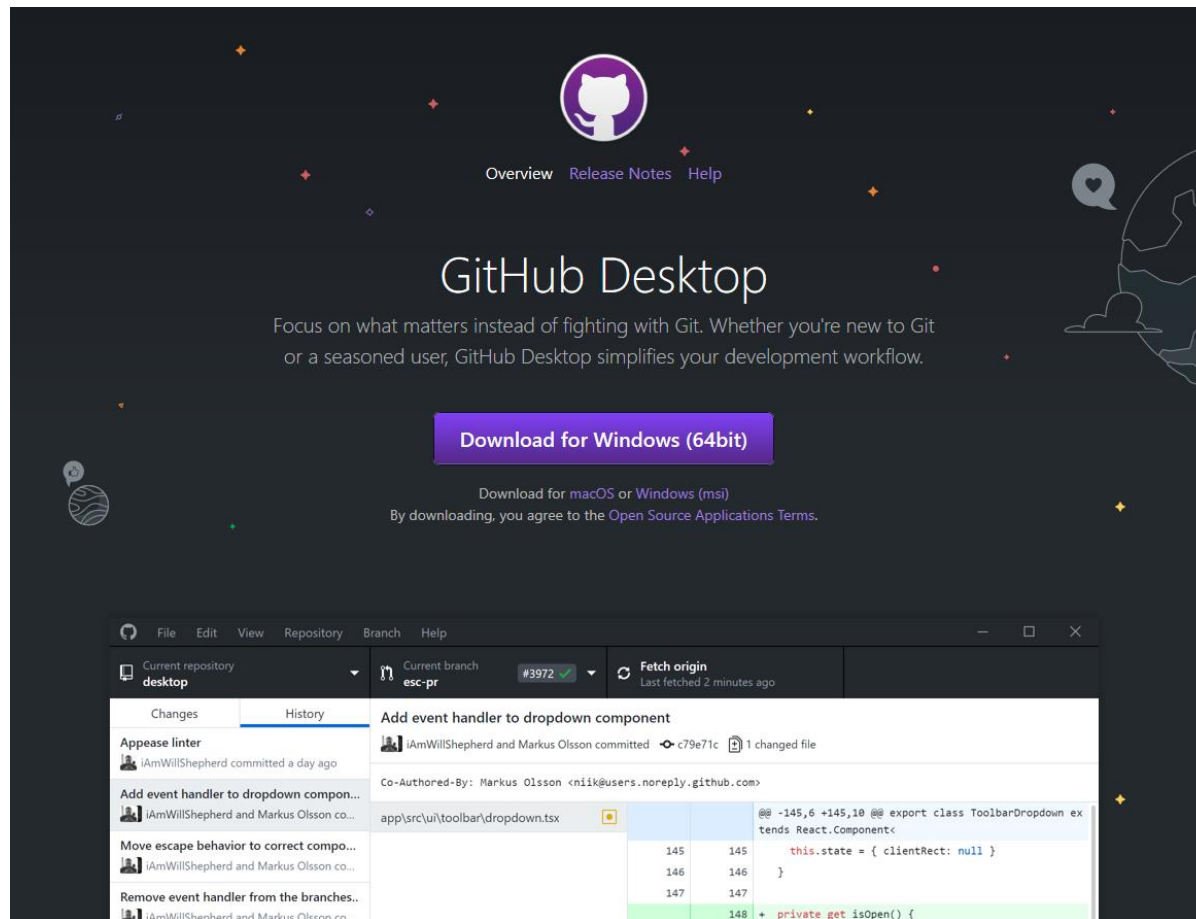
깃허브 계정 만들기
와
깃허브 데스크탑 설치

깃허브 계정 생성

- 마이크로소프트 사의 outlook 메일이나 구글 메일 주소를 생성
 - 마이크로소프트 사의 outlook 메일 계정 생성
 - MS 원드라이브 계정과 동일
- 깃허브 계정 생성
 - 위에서 만든 계정으로 깃허브 계정 생성
- 깃허브 로그인 성공

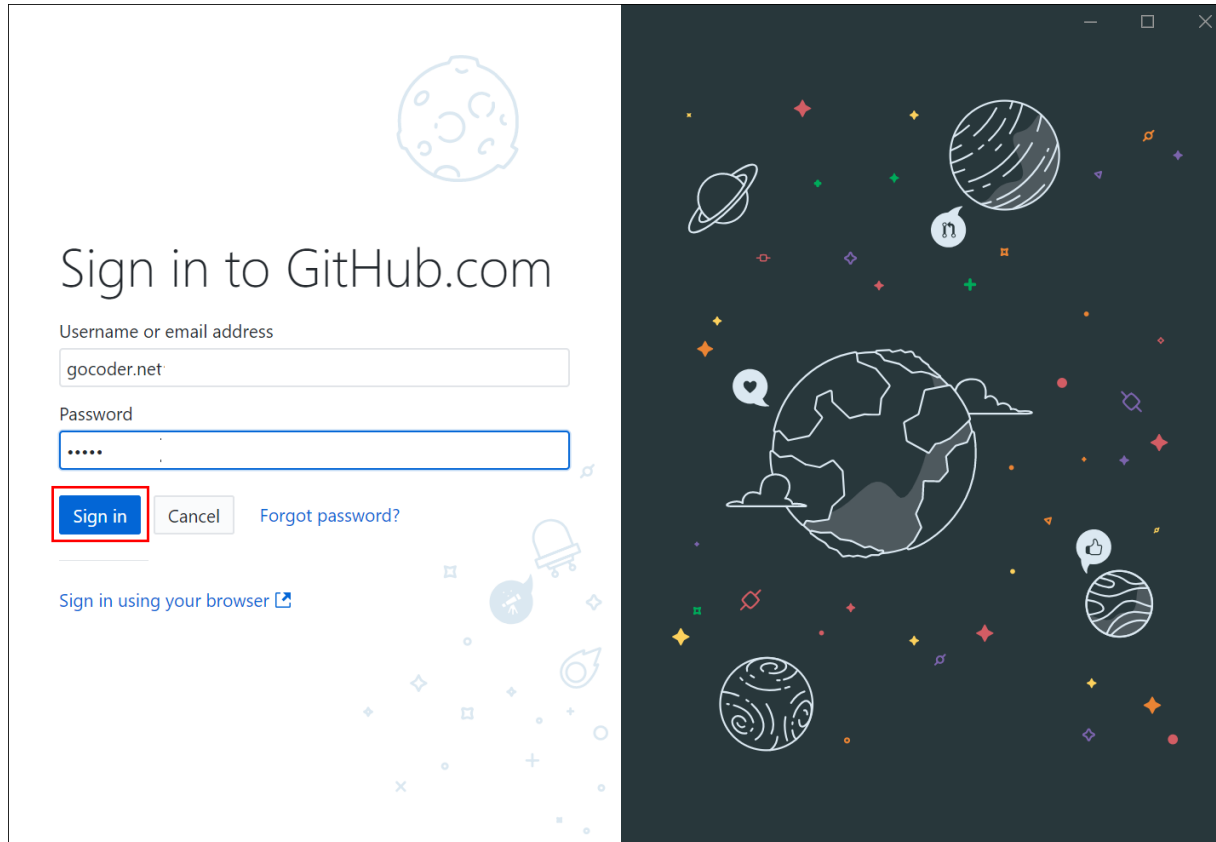
깃허브 데스크탑 설치

- <https://desktop.github.com/>



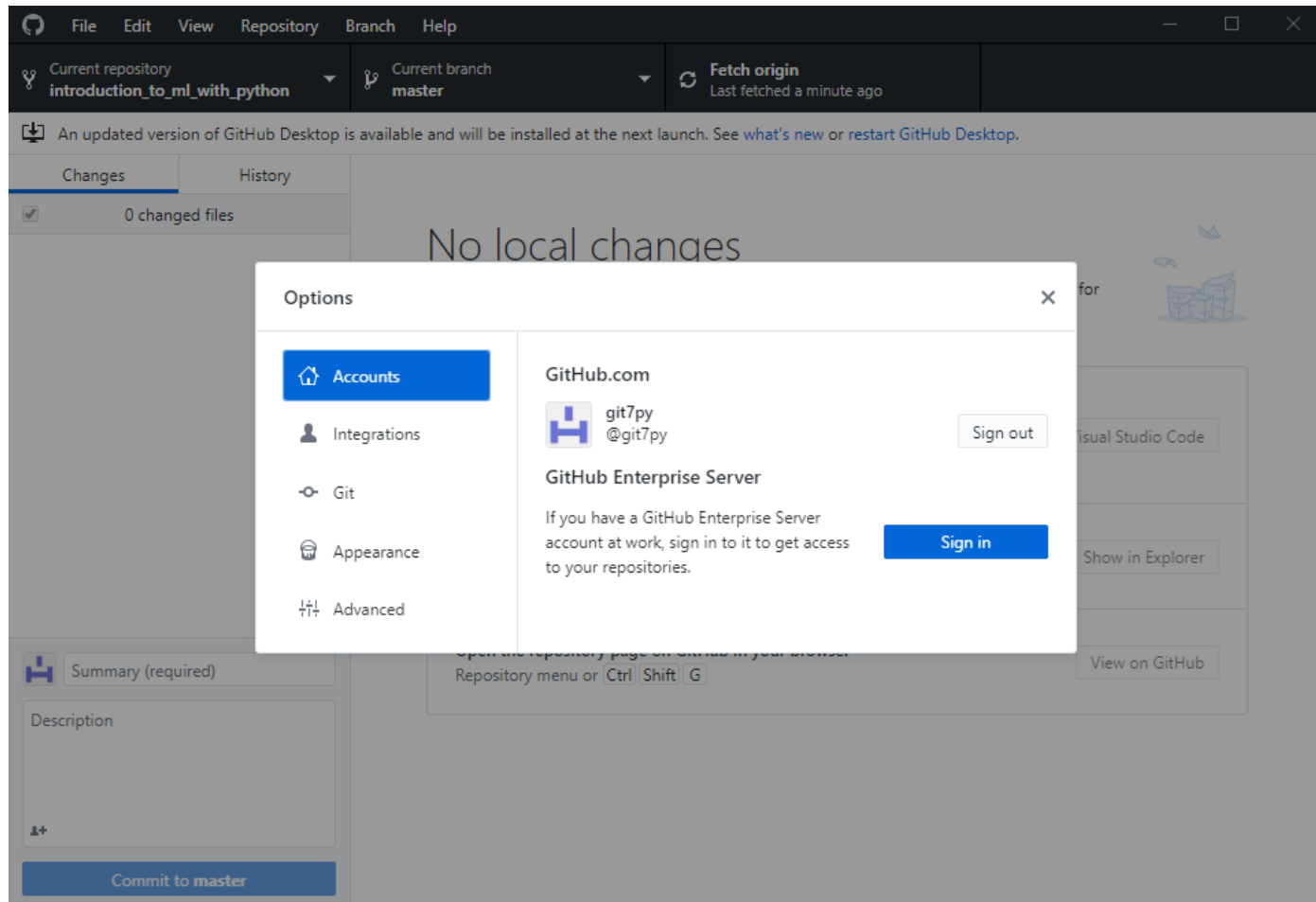
깃허브 로그인

- 중간에 자신의 메일로 인증 과정이 필요할 수도



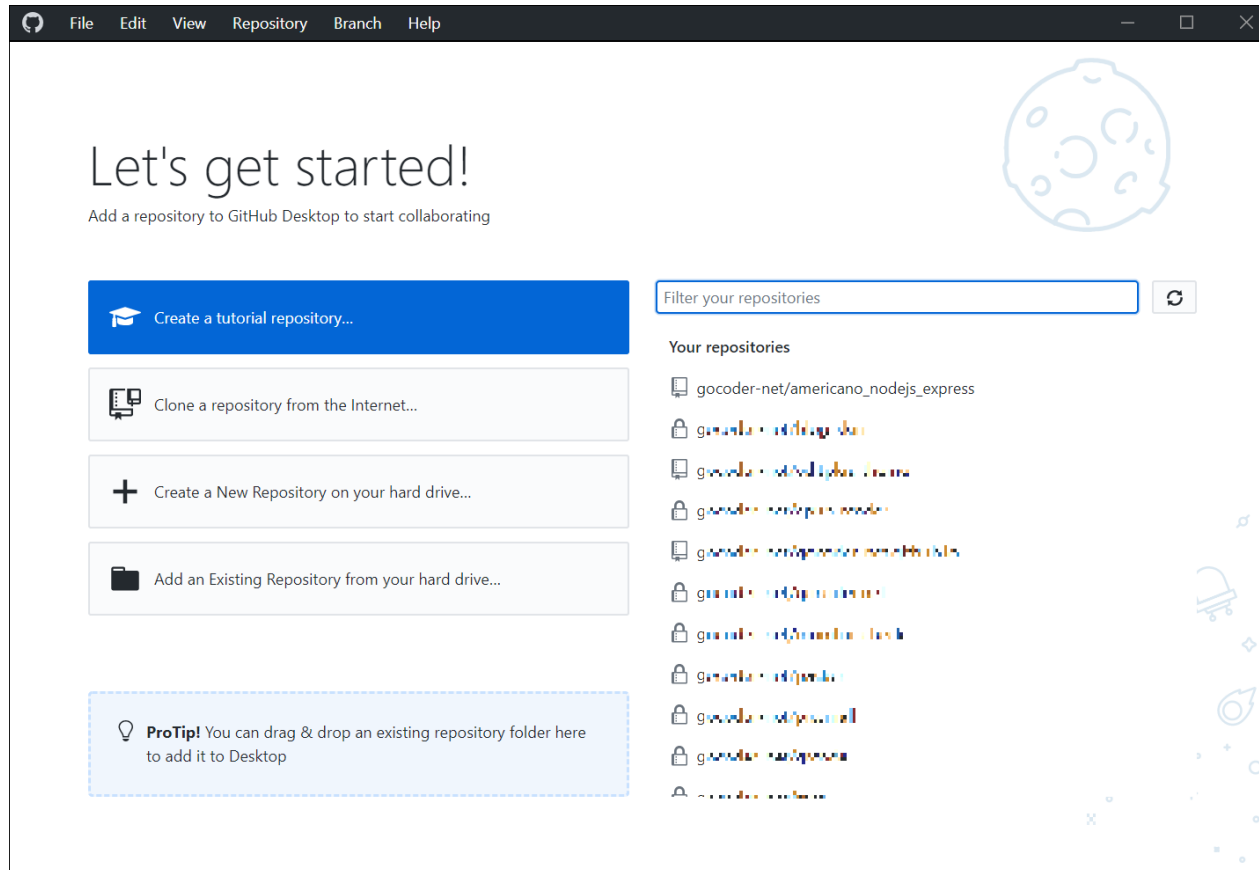
설치 후 기본 설정 확인

- 자신의 계정 id / 암호 로그인



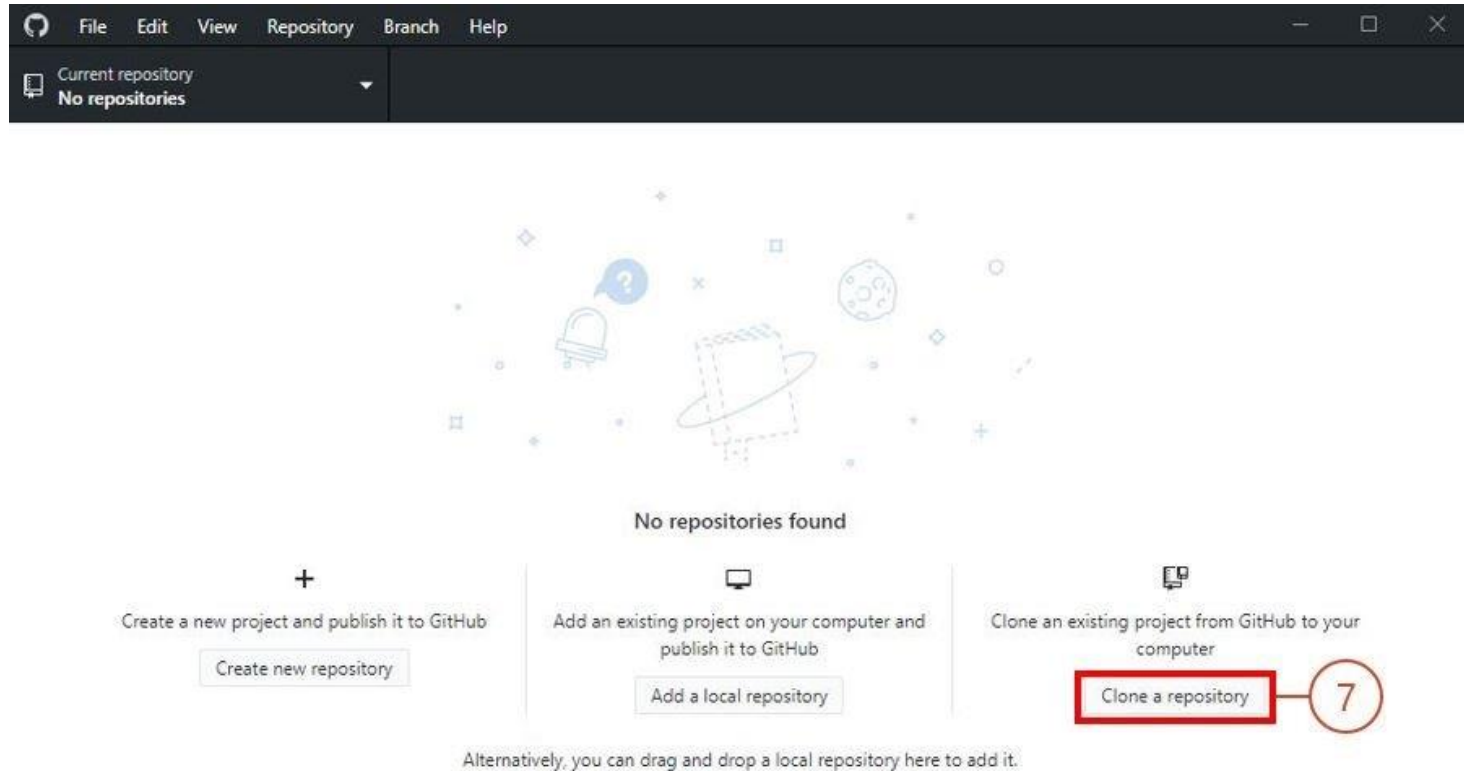
첫 화면

- Clone a repository from the internet...



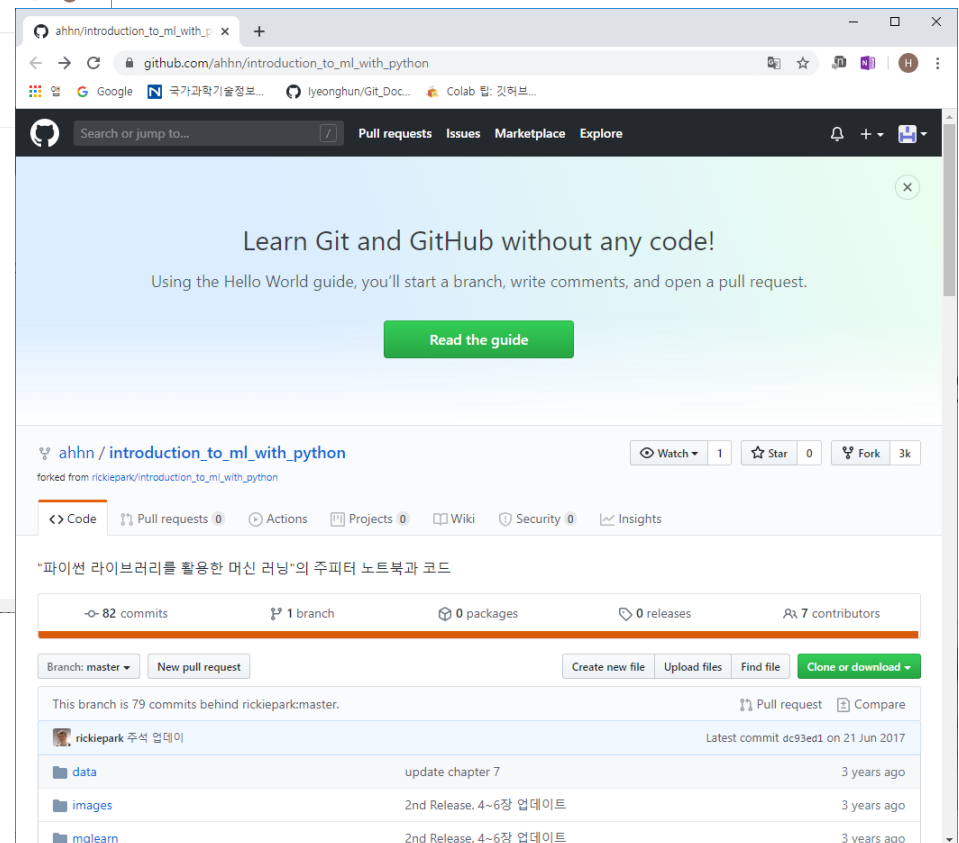
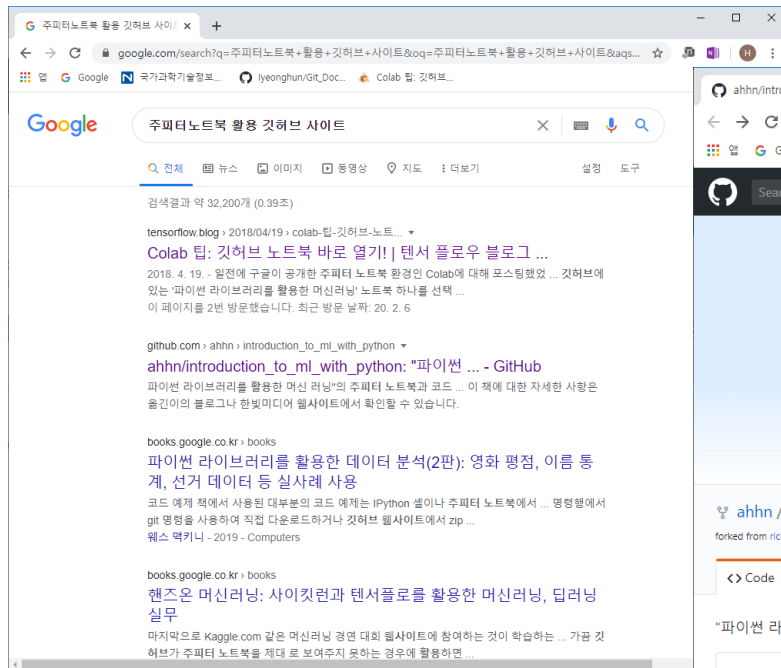
원격 저장소 복사(clone) 활용

Clone a repository



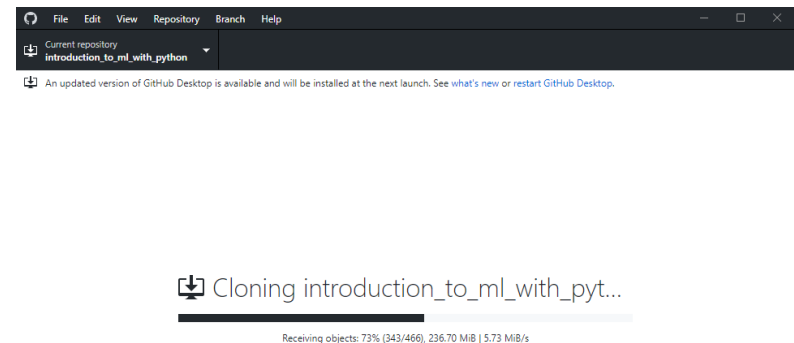
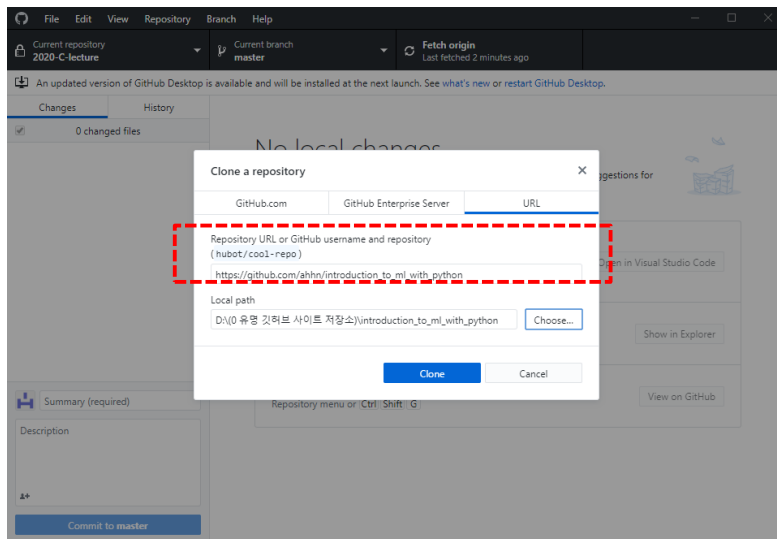
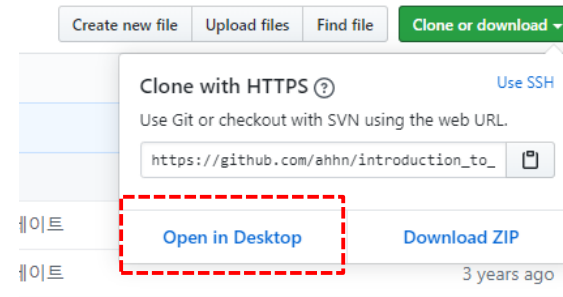
복사할 깃허브 사이트 검색

- 주피터노트북 활용 깃허브 사이트
 - https://github.com/ahhn/introduction_to_ml_with_python



Desktop으로 열기

- **Open in Desktop**
 - 깃허브 데스크탑 설치 필요
- **적당한 지역 저장소 선택**
 - Clone
- **저장소 복사**
 - 폴더 '2020 이공계 연수' 하부에 깃허브를 복사

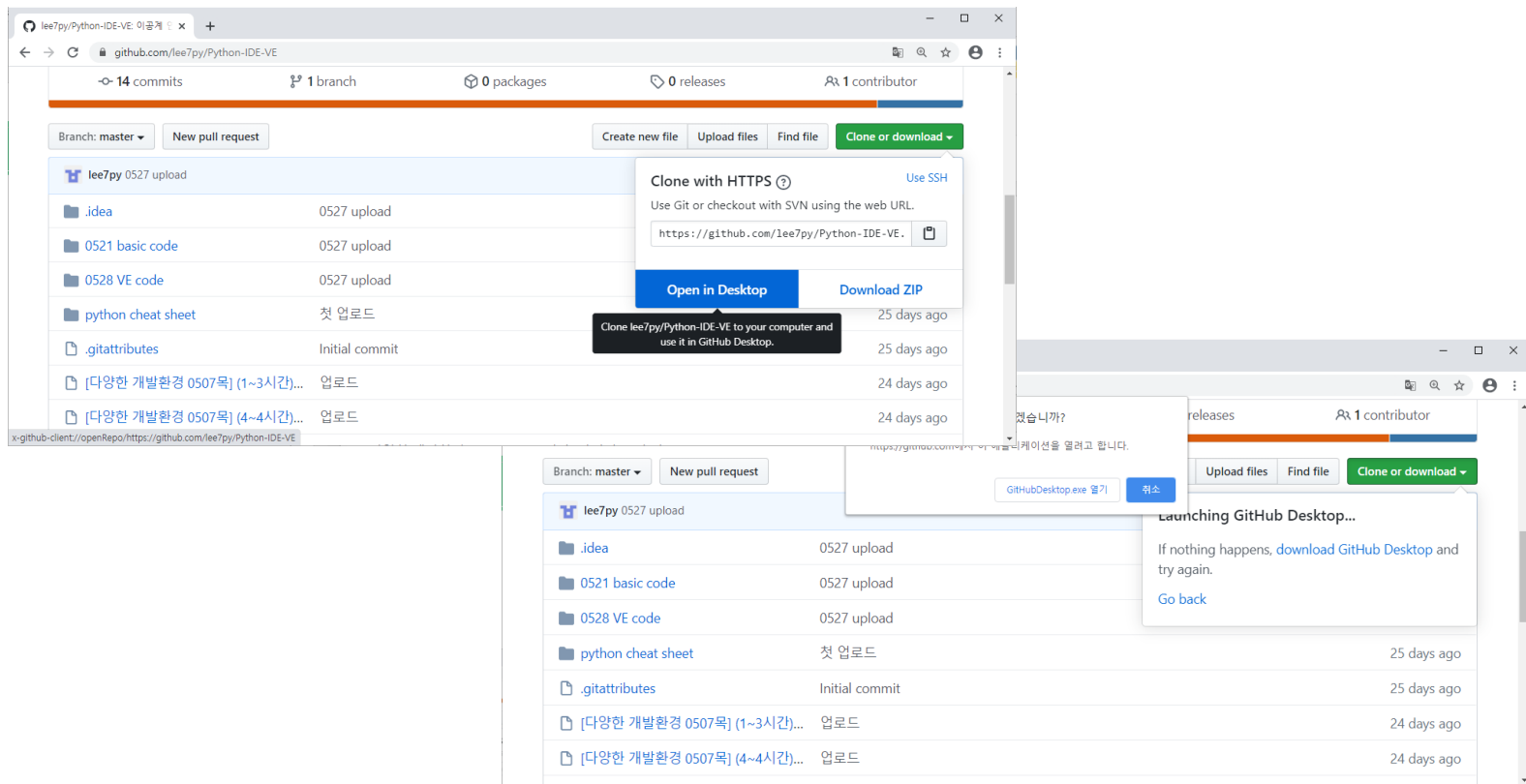


우리 수업 사이트 복사(1)

• 저장소 복사

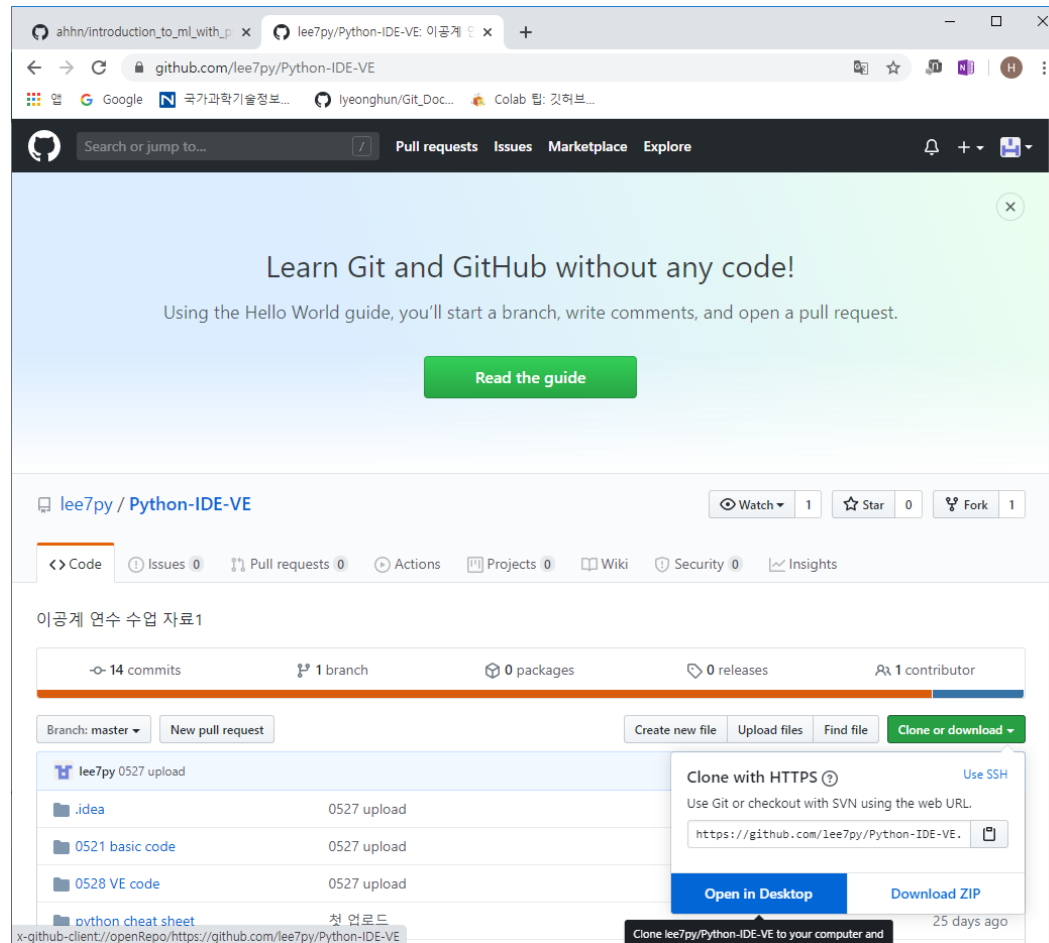
– 폴더 '2020 이공계 연수' 하부에 깃허브를 복사

• Python-IDE-VE



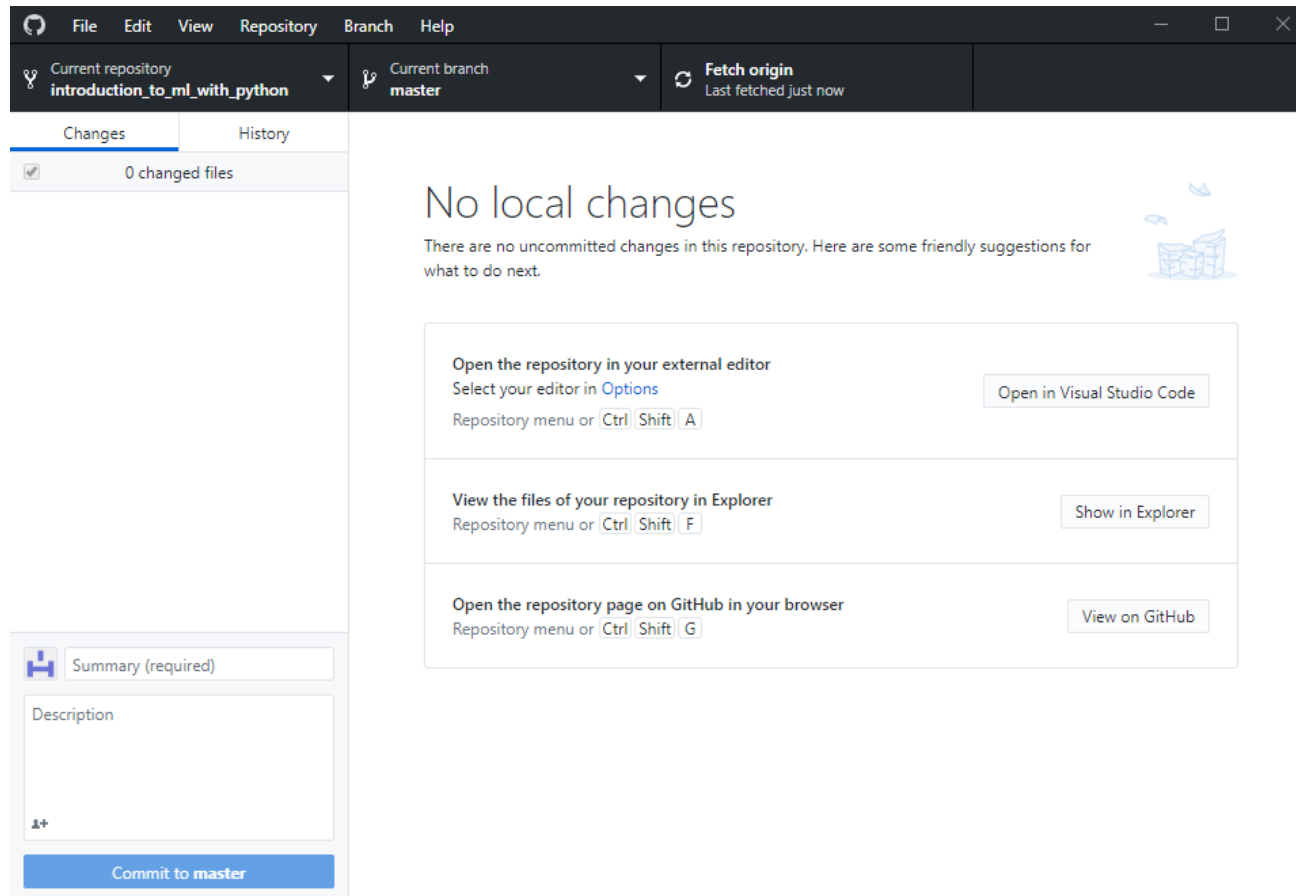
우리 수업 사이트 복사(2)

- 사이트 [github.com/lee7py](https://github.com/lee7py/Python-IDE-VE)에서
 - 수업 저장소 선택



다음 주에 점검

- 다음 화면에서



주피터 노트북 모듈로 설치하기

pip install notebook

• 다음도 가능

- pip install jupyter
- Jupyter로 설치하면 약간 더 많이 설치

```
C:\Windows\System32\cmd.exe
ERROR: unknown command "insatall" - maybe you meant "install"

D:\Python38-32>pip install notebook
Collecting notebook
  Downloading https://files.pythonhosted.org/packages/b1/f1/0a67f09ef53a342403ffa66646ee39273e0ac79ffa5de5dbe2f3e28b5bdf/notebook-6.0.3-py3-none-any.whl (9.7MB)
    [Progress bar] 9.7MB 3.2MB/s
Collecting jupyter-core>=4.6.1
  Using cached https://files.pythonhosted.org/packages/fb/82/86437f661875e30682e99d04c13ba6c216f86f5f6ca6ef212d3ee8b6ca11/jupyter_core-4.6.1-py2.py3-none-any.whl
Collecting Jinja2
  Using cached https://files.pythonhosted.org/packages/65/e0/eb35e762802015cab1ccee04e8a277b03f1d8e53da3ec3106882ec42558b/Jinja2-2.10.3-py2.py3-none-any.whl
Collecting jupyter-client>=5.3.4
  Using cached https://files.pythonhosted.org/packages/13/81/fe0ee1bcf949851a120254b1f530ae1e01bdde2d3ab9710c6ff81525061/jupyter_client-5.3.4-py2.py3-none-any.whl
Collecting ipython-genutils
  Downloading https://files.pythonhosted.org/packages/fa/bc/9bd3b5c2b4774d5f33b2d544f1460be9df7df2fe42f352135381c347c69a/ipython_genutils-0.2.0-py2.py3-none-any.whl
Collecting nbconvert
  Downloading https://files.pythonhosted.org/packages/79/6c/05a569e9f703d18aacb89b7ad6075b404e8a4afde2c26b73ca77bb644b14/nbconvert-5.6.1-py2.py3-none-any.whl (455kB)
    [Progress bar] 460kB 6.8MB/s
Collecting prometheus-client
  Downloading https://files.pythonhosted.org/packages/b3/23/41a5a24b502d35a4ad50a5bb7202a5e1d9a0364d0c12f56db3dbf7aca76d/prometheus_client-0.7.1.tar.gz
Collecting terminado>=0.8.1
```

```
C:\Windows\System32\cmd.exe
D:\Python38-32>pip list
Package Version
-----
anaconda 0.0.1.1
attrs 19.3.0
backcall 0.1.0
bleach 3.1.0
certifi 2019.11.28
colorama 0.4.3
decorator 4.4.1
defusedxml 0.6.0
entrypoints 0.3
ipykernel 5.1.3
ipython 7.11.1
ipython-genutils 0.2.0
jedi 0.15.2
Jinja2 2.10.3
jsonschema 3.2.0
jupyter-client 5.3.4
jupyter-core 4.6.1
MarkupSafe 1.1.1
mistune 0.8.4
nbconvert 5.6.1
nbformat 5.0.4
notebook 6.0.3
numpy 1.18.1
pandas 0.25.3
```

실습1: 가상환경을 만들어 노트북 설치 후 실행

- Virtualenv로 가상환경 nb 생성

Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

```
D:\myVE>virtualenv nb
Using base prefix 'c:\python\python37-32'
New python executable in D:\myVE\nb\Scripts\python.exe
Installing setuptools, pip, wheel...
done.
```

```
D:\myVE>nb\scripts\activate
```

```
(nb) D:\myVE>
```

실습1: 노트북 설치

(nb) D:\myVE> **pip install notebook**

Collecting notebook

Downloading notebook-6.0.3-py3-none-any.whl (9.7 MB)

 9.7 MB 504 kB/s

Collecting terminado>=0.8.1

Downloading terminado-0.8.3-py2.py3-none-any.whl (33 kB)

Collecting traitlets>=4.2.1

Downloading traitlets-4.3.3-py2.py3-none-any.whl (75 kB)

 75 kB ...

...

Successfully built prometheus-client pandocfilters pyrsistent backcall

Installing collected packages: pywinpty, tornado, terminado, decorator, ipython-genutils, six, traitlets, pyzmq, pywin32, jupyter-core, python-dateutil, jupyter-client, MarkupSafe, jinja2, more-itertools, zipp, importlib-metadata, pyrsistent, attrs, jsonschema, nbformat, Send2Trash, pygments, parso, jedi, backcall, colorama, wcwidth, prompt-toolkit, pickleshare, ipython, ipykernel, pandocfilters, testpath, entrypoints, mistune, defusedxml, webencodings, bleach, nbconvert, prometheus-client, notebook

Successfully installed MarkupSafe-1.1.1 Send2Trash-1.5.0 attrs-19.3.0 backcall-0.1.0 bleach-3.1.0 colorama-0.4.3 decorator-4.4.1 defusedxml-0.6.0 entrypoints-0.3 importlib-metadata-1.4.0 ipykernel-5.1.3 ipython-7.11.1 ipython-genutils-0.2.0 jedi-0.15.2 jinja2-2.10.3 jsonschema-3.2.0 jupyter-client-5.3.4 jupyter-core-4.6.1 mistune-0.8.4 more-itertools-8.1.0 nbconvert-5.6.1 nbformat-5.0.4 notebook-6.0.3 pandocfilters-1.4.2 parso-0.5.2 pickleshare-0.7.5 prometheus-client-0.7.1 prompt-toolkit-3.0.2 pygments-2.5.2 pyrsistent-0.15.7 python-dateutil-2.8.1 pywin32-227 pywinpty-0.5.7 pyzmq-18.1.1 six-1.14.0 terminado-0.8.3 testpath-0.4.4 tornado-6.0.3 traitlets-4.3.3 wcwidth-0.1.8 webencodings-0.5.1 zipp-2.0.1

(nb) D:\myVE>

실습1: 노트북 실행 1

(nb) D:\myVE> `python -m notebook`

[I 02:32:12.577 NotebookApp] The port 8888 is already in use, trying another port.

[I 02:32:12.578 NotebookApp] The port 8889 is already in use, trying another port.

[I 02:32:12.592 NotebookApp] Serving notebooks from local directory: D:\myVE

[I 02:32:12.592 NotebookApp] The Jupyter Notebook is running at:

[I 02:32:12.593 NotebookApp]

<http://localhost:8890/?token=93225b8c2a78a4dcfdac04c7066981153f50037ef153501c>

[I 02:32:12.594 NotebookApp] or

<http://127.0.0.1:8890/?token=93225b8c2a78a4dcfdac04c7066981153f50037ef153501c>

[I 02:32:12.594 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[C 02:32:12.863 NotebookApp]

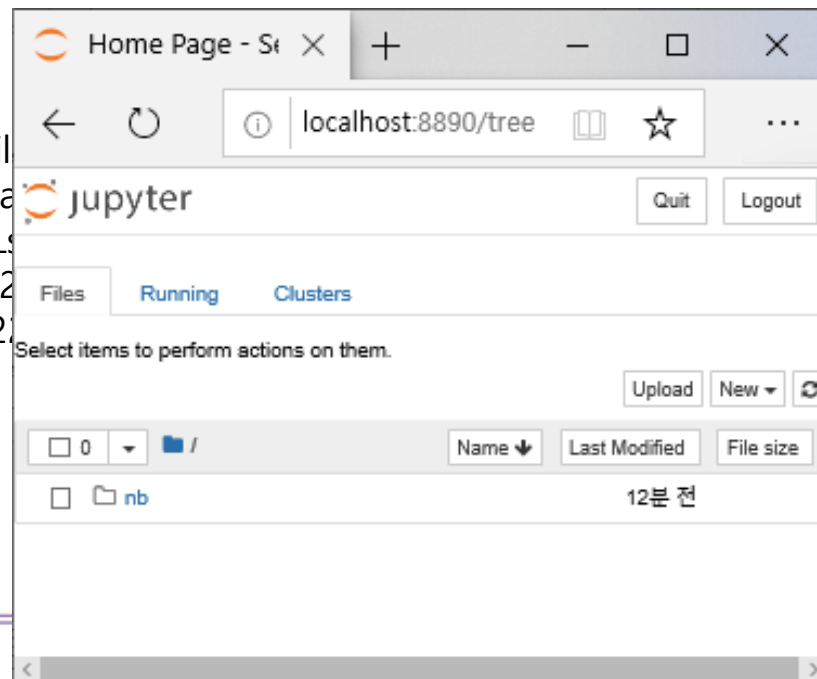
To access the notebook, open this file

`file:///C:/Users/USER/AppData/Roaming/jupyter/notebooks/`

Or copy and paste one of these URLs

<http://localhost:8890/?token=93225b8c2a78a4dcfdac04c7066981153f50037ef153501c>

or <http://127.0.0.1:8890/?token=93225b8c2a78a4dcfdac04c7066981153f50037ef153501c>



nl

01c

501c

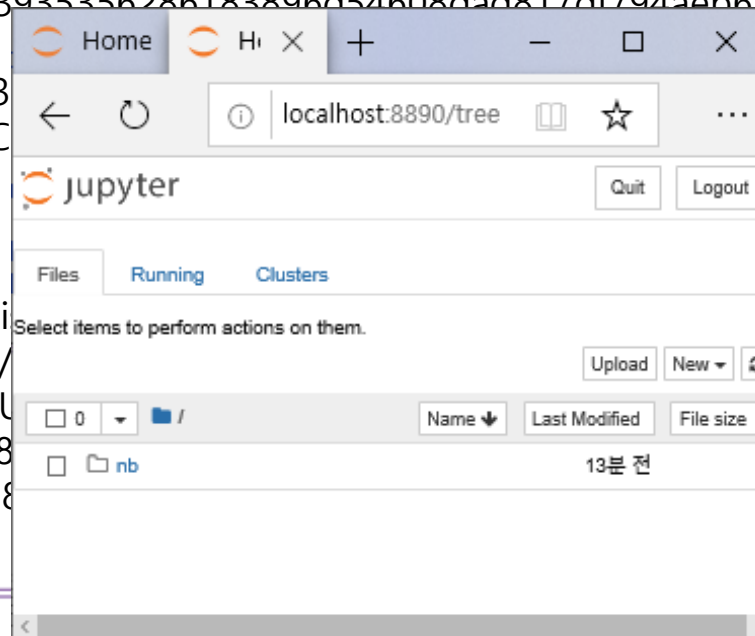
실습1: 노트북 실행 2

```
(nb) D:\myVE>where jupyter
D:\myVE\nb\Scripts\jupyter.exe
C:\Python\Anaconda3\Scripts\jupyter.exe
```

```
(nb) D:\myVE>jupyter notebook
```

```
[I 02:35:53.067 NotebookApp] The port 8888 is already in use, trying another port.
[I 02:35:53.068 NotebookApp] The port 8889 is already in use, trying another port.
[I 02:35:53.082 NotebookApp] Serving notebooks from local directory: D:\myVE
[I 02:35:53.082 NotebookApp] The Jupyter Notebook is running at:
[I 02:35:53.083 NotebookApp]
http://localhost:8890/?token=895033935356286183896d54608dad817df794aeb6a1d045
[I 02:35:53.084 NotebookApp] or
http://127.0.0.1:8890/?token=895033935356286183896d54608dad817df794aeb6a1d045
[I 02:35:53.084 NotebookApp] Use Ctrl-C to stop the application.
(twice to skip confirmation).
[C 02:35:53.331 NotebookApp]
```

To access the notebook, open this file:///C:/Users/USER/AppData/Local/Programs/Jupyter Notebook/
 Or copy and paste one of these URLs:
 http://localhost:8890/?token=895033935356286183896d54608dad817df794aeb6a1d045
 or http://127.0.0.1:8890/?token=895033935356286183896d54608dad817df794aeb6a1d045



실습2: 가상환경을 만들어 노트북 설치 후 실행

- **venv로 가상환경 nblab 생성**

Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

```
D:\myVE>python -m venv nblab
```

```
D:\myVE>nblab\scripts\activate
```

```
(nblab) D:\myVE>
```

27

실습2: 노트북 실행 1

```
(nblab) D:\WmyVE>python -m notebook
```

```
[I 03:00:44.517 NotebookApp] The port 8888 is already in use, trying another port.
```

```
[I 03:00:44.518 NotebookApp] The port 8889 is already in use, trying another port.
```

```
[I 03:00:44.529 NotebookApp] Serving notebooks from local directory: D:\WmyVE
```

```
[I 03:00:44.529 NotebookApp] The Jupyter Notebook is running at:
```

```
[I 03:00:44.529 NotebookApp]
```

```
http://localhost:8890/?token=9e39780498cecdccbd61fad416d6ab471d6a7bf797ab2ca
```

```
[I 03:00:44.529 NotebookApp] or
```

```
http://127.0.0.1:8890/?token=9e39780498cecdccbd61fad416d6ab471d6a7bf797ab2ca
```

```
[I 03:00:44.530 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

```
[C 03:00:44.786 NotebookApp]
```

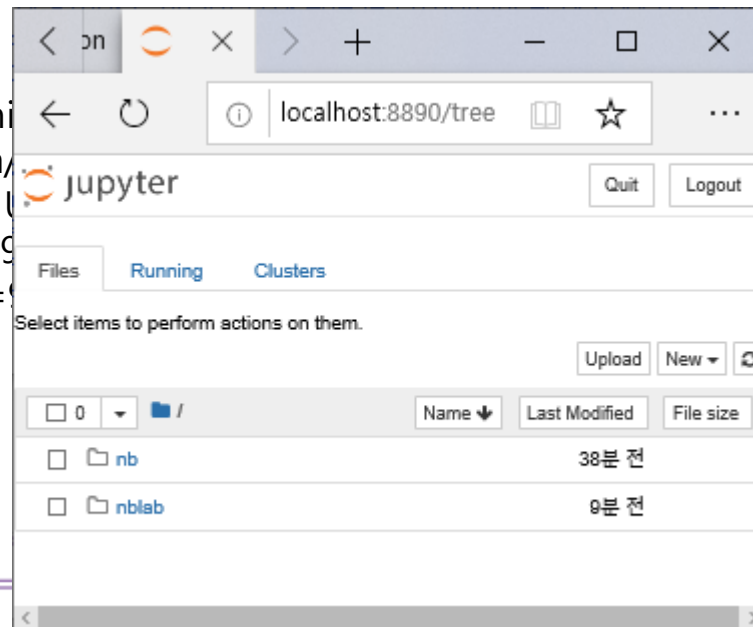
To access the notebook, open the

file:///C:/Users/USER/AppData/

Or copy and paste one of these URLs:

http://localhost:8890/?token=9e39780498cecdccbd61fad416d6ab471d6a7bf797ab2ca

or http://127.0.0.1:8890/?token=9e39780498cecdccbd61fad416d6ab471d6a7bf797ab2ca



open.html

bf797ab2ca

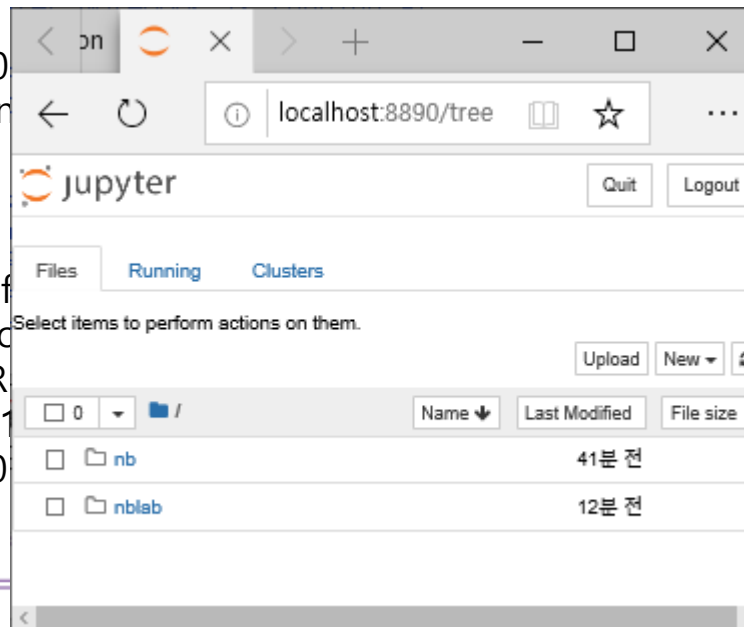
bf797ab2ca

실습2: 노트북 실행 2

```
(nblab) D:\myVE>where jupyter
D:\myVE\nblab\Scripts\jupyter.exe
C:\Python\Anaconda3\Scripts\jupyter.exe
```

```
(nblab) D:\myVE>jupyter notebook
[I 03:03:37.866 NotebookApp] The port 8888 is already in use, trying another port.
[I 03:03:37.867 NotebookApp] The port 8889 is already in use, trying another port.
[I 03:03:37.879 NotebookApp] Serving notebooks from local directory: D:\myVE
[I 03:03:37.879 NotebookApp] The Jupyter Notebook is running at:
[I 03:03:37.880 NotebookApp]
http://localhost:8890/?token=b010caa00636b46607257b9eb3157b0be0aec56679054009
[I 03:03:37.881 NotebookApp] or
http://127.0.0.1:8890/?token=b010caa00636b46607257b9eb3157b0be0aec56679054009
[I 03:03:37.881 NotebookApp] Use Ctrl-C to stop the server.
[C 03:03:38.147 NotebookApp]
```

To access the notebook, open this file:///C:/Users/USER/AppData/Roaming/Python/Python36/Jupyter/notebook.html
Or copy and paste one of these URLs:
http://localhost:8890/?token=b010caa00636b46607257b9eb3157b0be0aec56679054009
or http://127.0.0.1:8890/?token=b010caa00636b46607257b9eb3157b0be0aec56679054009



모듈 notebook, jupyter 설치 비교

(nb) D:\WmyVE>pip freeze

```
attrs==19.3.0
backcall==0.1.0
bleach==3.1.0
colorama==0.4.3
decorator==4.4.1
defusedxml==0.6.0
entrypoints==0.3
importlib-metadata==1.4.0
ipykernel==5.1.3
ipython==7.11.1
ipython-genutils==0.2.0
jedi==0.15.2
Jinja2==2.10.3
jsonschema==3.2.0
jupyter-client==5.3.4
jupyter-core==4.6.1
MarkupSafe==1.1.1
mistune==0.8.4
more-itertools==8.1.0
nbconvert==5.6.1
nbformat==5.0.4
notebook==6.0.3
pandocfilters==1.4.2
parso==0.5.2
```

```
pickleshare==0.7.5
prometheus-
client==0.7.1
prompt-toolkit==3.0.2
Pygments==2.5.2
pysistent==0.15.7
python-dateutil==2.8.1
pywin32==227
pywinpty==0.5.7
pyzmq==18.1.1
Send2Trash==1.5.0
six==1.14.0
terminado==0.8.3
testpath==0.4.4
tornado==6.0.3
traitlets==4.3.3
wcwidth==0.1.8
webencodings==0.5.1
zipp==2.0.1
```

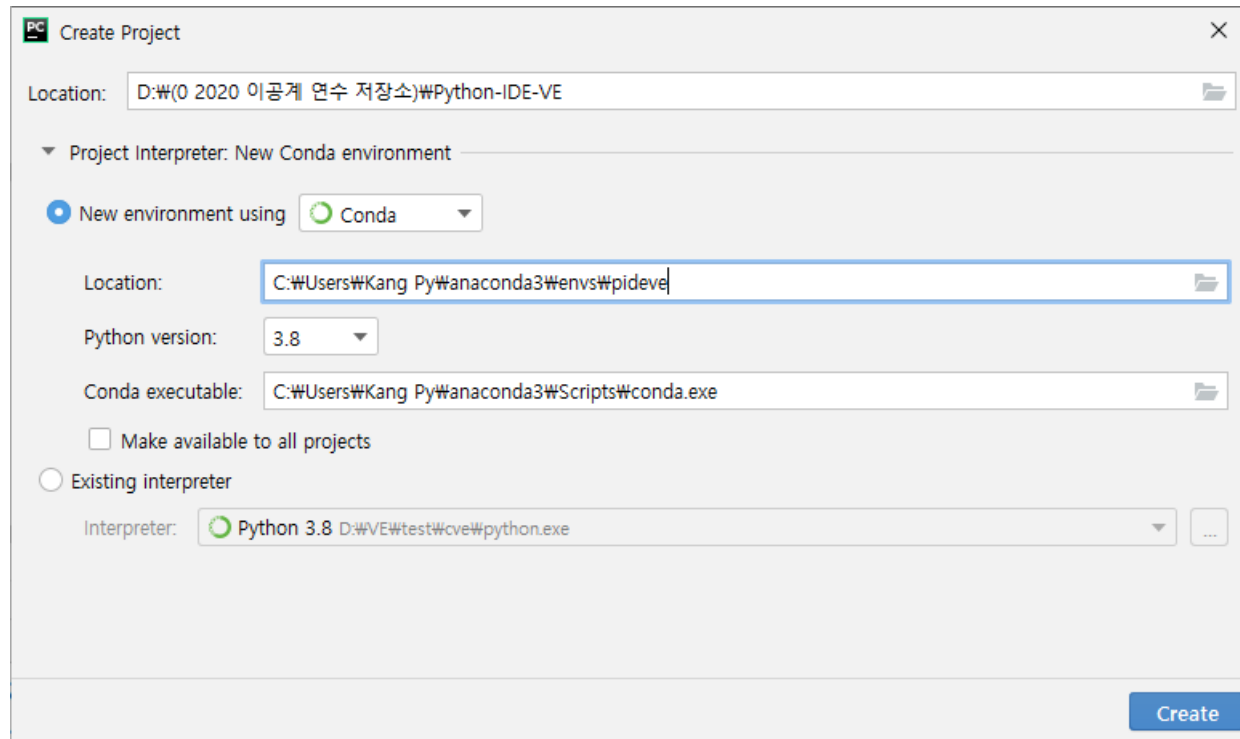
(nblab) D:\WmyVE>pip freeze

```
attrs==19.3.0
backcall==0.1.0
bleach==3.1.0
colorama==0.4.3
decorator==4.4.1
defusedxml==0.6.0
entrypoints==0.3
importlib-metadata==1.4.0
ipykernel==5.1.3
ipython==7.11.1
ipython-genutils==0.2.0
ipywidgets==7.5.1
jedi==0.15.2
Jinja2==2.10.3
jsonschema==3.2.0
jupyter==1.0.0
jupyter-client==5.3.4
jupyter-console==6.1.0
jupyter-core==4.6.1
MarkupSafe==1.1.1
mistune==0.8.4
more-itertools==8.1.0
nbconvert==5.6.1
nbformat==5.0.4
```

```
notebook==6.0.3
pandocfilters==1.4.2
parso==0.5.2
pickleshare==0.7.5
prometheus-client==0.7.1
prompt-toolkit==3.0.2
Pygments==2.5.2
pysistent==0.15.7
python-dateutil==2.8.1
pywin32==227
pywinpty==0.5.7
pyzmq==18.1.1
qtconsole==4.6.0
Send2Trash==1.5.0
six==1.14.0
terminado==0.8.3
testpath==0.4.4
tornado==6.0.3
traitlets==4.3.3
wcwidth==0.1.8
webencodings==0.5.1
widgetsnextension==3.5.1
zipp==2.0.1
```

파이참 개발환경 설정

프로젝트 생성



프로젝트 완성

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The title bar indicates the current project is 'Python-IDE-VE' located at 'D:\(0 2020 이공계 연수 저장소)\Python-IDE-VE'.

The left sidebar shows the Project structure for 'Python-IDE-VE'. The files and folders listed are:

- 0521 basic code
- 0528 VE code
- 0604 jupyter code
 - .ipynb_checkpoints
 - jupyterex1.ipynb
 - mdtest.py
- python cheat sheet
 - .gitattributes
 - [다양한 개발환경 0507목] (1~3시간) 파이썬의 기본 개발도구 python.org.pdf
 - [다양한 개발환경 0507목] (4~4시간) 캐글 활용.pdf
 - [다양한 개발환경 0507목] (5~6시간) 파이썬 문법 기본.pdf
 - [다양한 개발환경 0514목] (1~2시간) 아나콘다 개요와 설치.pdf
 - [다양한 개발환경 0514목] (3~4시간) 주피터 노트북 개요와 활용 기초.pdf
 - [다양한 개발환경 0514목] (5~6시간) 파이썬 문법 기본(리스트, 딕셔너리 중심).pdf

The bottom panel shows a terminal window with the following output:

```
(pideve) D:\(0 2020 이공계 연수 저장소)\Python-IDE-VE>conda list
# packages in environment at C:\Users\Kang Py\anaconda3\envs\pideve:
#
# Name                          Version                      Build      Channel
ca-certificates                 2020.1.1                     0
certifi                         2020.4.5.1                   py38_0
openssl                         1.1.1g                       he774522_0
pip                             20.0.2                       py38_3
python                          3.8.3                        he1778fa_0
setuptools                     46.4.0                       py38_0
sqlite                          3.31.1                       h2a8f88b_1
vc                              14.1                         h0510ff6_4
vs2015_runtime                 14.16.27012                  hf0eaf9b_2
wheel                           0.34.2                       py38_0
wincertstore                   0.2                          py38_0
```

Jupyter 및 matplotlib 설치

- Conda install notebook matplotlib pandas bokeh

The screenshot shows a PyCharm IDE window titled 'Python-IDE-VE [D:\(0 2020 이공계 연수 저장소)\Python-IDE-VE] - PyCharm'. The left sidebar displays the project structure, including folders for '0521 basic code', '0528 VE code', '0604 jupyter code', and 'python cheat sheet'. The main editor area is currently blank. At the bottom, a terminal window is open, showing the following commands and output:

```

done

(pideve) D:\(0 2020 이공계 연수 저장소)\Python-IDE-VE>conda list jupyter
# packages in environment at C:\Users\Kang Py\anaconda3\envs\pideve:
#
# Name                Version             Build Channel
jupyter_client         6.1.3               py_0
jupyter_core           4.6.3               py38_0

(pideve) D:\(0 2020 이공계 연수 저장소)\Python-IDE-VE>conda list notebook
# packages in environment at C:\Users\Kang Py\anaconda3\envs\pideve:
#
# Name                Version             Build Channel
notebook              6.0.3               py38_0

(pideve) D:\(0 2020 이공계 연수 저장소)\Python-IDE-VE>
  
```

The terminal output confirms the successful installation of Jupyter (jupyter_client 6.1.3, jupyter_core 4.6.3) and Notebook (notebook 6.0.3) in the specified environment.

주피터 노트북 설치와 실행

- 설치 2가지
 - conda install jupyter
 - conda install notebook
- 실행 2가지
 - jupyter notebook
 - python -m notebook

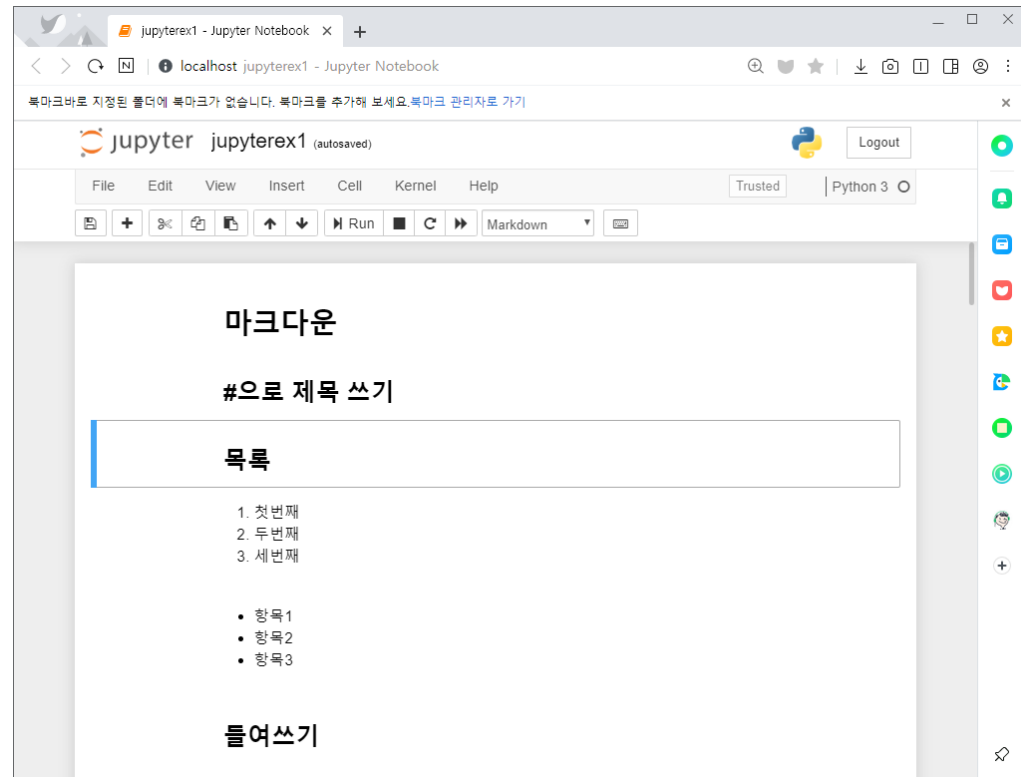
```
(pideve) D:\(0 2020 이공계 연수 저장소)\Python-IDE-VE>python -m notebook
[I 15:33:32.161 NotebookApp] The port 8888 is already in use, trying another port.
[I 15:33:32.162 NotebookApp] The port 8889 is already in use, trying another port.
[I 15:33:32.172 NotebookApp] Serving notebooks from local directory: D:\(0 2020 이공계 연수 저장소)\Python-IDE-VE
[I 15:33:32.172 NotebookApp] The Jupyter Notebook is running at:
[I 15:33:32.173 NotebookApp] http://localhost:8890/?token=f17bc3176370ba801958df09b1decf2322886b8b07700edc
[I 15:33:32.173 NotebookApp] or http://127.0.0.1:8890/?token=f17bc3176370ba801958df09b1decf2322886b8b07700edc
[I 15:33:32.174 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 15:33:32.238 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/Kang%20Py/AppData/Roaming/jupyter/runtime/nbserver-60092-open.html
```

주피터 노트북
마크다운 편집

실습 파일

- 주피터 노트북 실행
 - 터미널 열기
 - `python -m notebook`
 - `jupyter notebook`
- 파일 열기
 - `jupyterex1.ipynb`



마크다운 개요

- 마크다운은 텍스트 기반의 마크업 언어로 html으로 변환이 가능
 - 마크다운에서도 6가지의 제목을 나타내는 문법이 있는데 #을 사용
 - 아래처럼 각각 개수만큼 각 숫자에 맞는 h 태그 <h1> 태그

```
# 첫 번째 큰 제목
## 두 번째 큰 제목
### 세 번째 큰 제목
#### 네 번째 제목
##### 다섯 번째 제목
##### 여섯 번째 제목
#####
```

제목

첫 번째 큰 제목

두 번째 큰 제목

세 번째 큰 제목

네 번째 제목

다섯 번째 제목

여섯 번째 제목

#####

실행 결과

목록

- 숫자와 함께 쓰면 순서가 있는 목록
- 글머리 기호와 함께 쓰면 순서가 없는 목록

```
1. 첫번째
2. 두번째
3. 세번째
```

```
+ 첫번째
+ 두번째
+ 세번째
```

```
- 첫번째
- 두번째
- 세번째
```

```
* 첫번째
* 두번째
* 세번째
```

```
1. 첫번째
2. 두번째
3. 세번째
```

```
• 첫번째
• 두번째
• 세번째
• 첫번째
• 두번째
• 세번째
• 첫번째
• 두번째
• 세번째
```

순서가 있는 목록, 순서가 없는 목록

순서가 있는 목록, 순서가 없는 목록 결과

들어 쓰기

- 목록 안의 목록을 쓰고 싶다면 tab을 한 번 해주고 사용

```
+ 첫째
  + 둘째
    + 셋째

- 첫째
  - 둘째
    - 셋째

* 첫째
  * 둘째
    * 셋째
```

tab을 각각 해주면 색이 바뀌는 것을 볼 수 있다.

- 첫째
 - 둘째
 - 셋째
- 첫째
 - 둘째
 - 셋째
- 첫째
 - 둘째
 - 셋째

tab에 따라 목록 기호가 바뀌어 있다.

수식 출력

- \$와 \$ 사이에 기술

$$y = 2n^2$$

```
1 $y=2n^2$
```

$$c = \sqrt{a^2 + b^2}$$

```
1 $c=#sqrt{a^2 + b^2}$
```

주피터 노트북
매직 명령어 %, %%

%로 시작하는 명령어

• 보통 %문자로 시작

- 만약 사용자 변수가 선언되지 않았다면 %문자 없이도 기능을 수행
- 예를 들어 cls라는 변수가 선언되어 있다면 %cls라고 입력해야 하지만

- cls라는 변수가 없다면 그냥 cls라고 입력해도 동작을 수행

| 매직명령어 | 기능 |
|---------------------------|--|
| %magic %lsmagic | 모든 매직 명령어의 도움말 출력 매직 명령어 리스트 |
| %automagic | 매직함수를 %없이도 실행하게끔 함(default) 또는 %를 붙여야만 실행하게끔 함(실행할 때마다 전환됨) |
| %pprint | pretty print 모드의 on/off 전환 |
| %exit %quit | 물어보지 않고 IPython을 종료시킨다. |
| %cls | 화면 클리어 |
| %who %who_ls %whos | 변수의 리스트를 보여준다. 변수 리스트를 파이썬 리스트로 반환한다. 변수명 뿐만 아니라 변수 값도 보여준다. |
| %reset %reset -f | 작업공간을 초기화 시킨다. 수행 여부를 묻지 않고 초기화 시킨다. |
| %run file.py | file.py 파일을 실행시킨다. |
| %paste | 클립보드의 코드를 실행한다. |
| %cpaste %edit (or %ed) | 코드 블록을 직접 입력한 후 실행한다. 텍스트에디터를 실행시킨 후 거기에 입력한 코드를 실행한다. |
| %time %timeit | 실행시간을 측정해서 보여준다. 여러 번 실행한 후 실행시간을 분석한다. |
| %hist | 과거 명령어 리스트(history) 출력 |

주요 명령어

| 명령어 | 설명 |
|--------------------|----------------------------------|
| %pwd, %cd | 현재 위치 및 다른 디렉토리로 이동 |
| %history | 명령어 히스토리 출력 |
| %reset | 모든 정의된 변수 삭제 |
| %%capture | 실행되는 명령에 대한 정보의 결과를 저장 |
| %whos | 현재 정의된 변수 표시 |
| %pdoc, %psource | Help 기능 실행 |
| %timeit | 평균 실행 시간을 출력 |
| %bookmark | 디렉토리에 대한 별칭을 저장하고 쉽게 이동할 수 있게 해줌 |
| %%writefile | 현재 디렉토리에 파일 생성 |
| %load | 디렉토리에 있는 파일을 셀에 로딩 |
| %run | py 프록램 파일을 실행 |
| %matplotlib inline | matplotlib을 내부 셀에서 실행하기 |

소스 파일 생성 및 실행

- %%매직명령어
 - 셀 전체에 영향, 셀 매직
 - %%writefile 파일이름
- %매직명령어
 - 한 줄의 명령, 라인 매직
 - %run 파일이름
- 다음으로 파이썬 *.py 코드
 - 편집과 실행이 가능

```
In [18]: 1 %%writefile rndtest.py
          2 import random as rd
          3
          4 print(rd.random())
          5 print(rd.randint(1, 3))
          6 print(rd.choice(range(10)))
```

Writing rndtest.py

```
In [19]: 1 %run rndtest.py

0.9395107979835833
2
3
```

소스파일 셀에 올려 실행 준비

- %load 파일명
- %loadpy 파일명

```
In [ ]: 1 %load rndtest.py
        2 %loadpy rndtest.py

In [49]: 1 # %load rndtest.py
        2 import random as rd
        3
        4 print(rd.random())
        5 print(rd.randint(1, 3))
        6 print(rd.choice(range(10)))
        7
0.8233356785829301
1
0
```

매직 명령어 전체 보기

- %lsmagic

```
In [32]: 1 %lsmagic
```

```
Out[32]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark
%cd %clear %cls %colors %conda %config %connect_info %copy %ddir %debug
%dhist %dirs %doctest_mode %echo %ed %edit %env %gui %hist %history
%killbgscripts %ldir %less %load %load_ext %loadpy %logoff %logon %logstart
%logstate %logstop %ls %lsmagic %macro %magic %matplotlib %mkdir %more
%notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %pip %popd
%pprint %precision %prun %psearch %psource %pushd %pwd %pycat %pylab
%qtconsole %quickref %recall %rehashx %reload_ext %ren %rep %rerun %reset
%reset_selective %rmdir %run %save %sc %set_env %store %sx %system %tb
%time %timeit %unalias %unload_ext %who %who_ls %whos %xdel %xmode
```

```
Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html
%%javascript %%js %%latex %%markdown %%perl %%prun %%pypy %%python
%%python2 %%python3 %%ruby %%script %%sh %%svg %%sx %%system
%%time %%timeit %%writefile
```

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

실행 시간

• %timeit

- ms(밀리/초: milli second) : 10^{-3}
- μ s(마이크로/초: micro second) : 10^{-6}
- ns(나노/초: nano second) : 10^{-9}
- ps(피코/초: pico second) : 10^{-12}
- fs(펨토/초: femto second) : 10^{-15}
- as(아토/초: atto second) : 10^{-18}

```
In [33]: 1 %timeit?
```

```
In [41]: 1 import random as rd
          2
          3 %timeit rd.choices(list(range(10000)), k = 100000)
```

22.1 ms \pm 190 μ s per loop (mean \pm std. dev. of 7 runs, 10 loops each)

Docstring:

Time execution of a Python statement or expression

Usage, in line mode:

```
%timeit [-n<N> -r<R> [-t|-c] -q -p<P> -o] statement
```

or in cell mode:

```
%%timeit [-n<N> -r<R> [-t|-c] -q -p<P> -o] setup_code
code
code...
```

Time execution of a Python statement or expression using the timeit module. This function can be used both as a line and cell magic:

- In line mode you can time a single-line statement (though multiple ones can be chained with using semicolons).
- In cell mode, the statement in the first line is used as setup code (executed but not timed) and the body of the cell is timed. The cell body has access to any variables created in the setup code.

변수 관리

- %whos
- %reset

```
In [41]: 1 import random as rd
          2
          3 %timeit rd.choices(list(range(10000)), k = 100000)

22.1 ms ± 190 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

```
In [42]: 1 %whos

Variable Type      Data/Info
-----
rd       module      <module 'random' from 'd:\>yve\lib\random.py'>
```

```
In [43]: 1 %reset
          2 %whos

Once deleted, variables cannot be recovered. Proceed (y/[n])? y
Interactive namespace is empty.
```

```
In [44]: 1 year = 2020
          2 s = 'python'
          3 n = list('java')
          4 %whos

Variable Type      Data/Info
-----
n       list       n=4
s       str        python
year    int        2020
```

이전 내용이 모두 출력

- %history

```
In [45]: 1 %history
          print('Hello World!')
          import random as rd

          print(rd.random())
          print(rd.randint(1, 3))
          import random as rd

          print(rd.random())
          print(rd.randint(1, 3))
          import random as rd

          print(rd.random())
          print(rd.randint(1, 3))
          print(rd.choice(range(10)))
          list?
```

레이텍 수식 활용

- **%%latex**
 - `%%[시작 %]` 로 종료

```
In [26]: 1 %%latex
          2 %%[x^n + y^n = z^n%%]
          
$$x^n + y^n = z^n$$

```

주피터 노트북
셸 명령어!

!로 시작하는 쉘 명령어

• 명령 프롬프트에서의 명령어

| shell | magic | 설명 |
|-----------|-----------|---------------------|
| !pwd, !cd | %pwd, %cd | 현재 위치 및 다른 디렉토리로 이동 |
| !env | %env | 컴퓨터 환경정보 보기 |
| !echo | %echo | 메시지 출력하기 |
| !cp | %cp | 카피하기 |
| !ls | %ls | 현재 디렉토리의 리스트 |
| !mkdir | %mkdir | 디렉토리 생성 |
| !rmdir | %rmdir | 디렉토리 삭제 |
| !mv | %mv | 파일 이동 |
| !rm | %rm | 파일 삭제 |