

# Python 패키지 설치

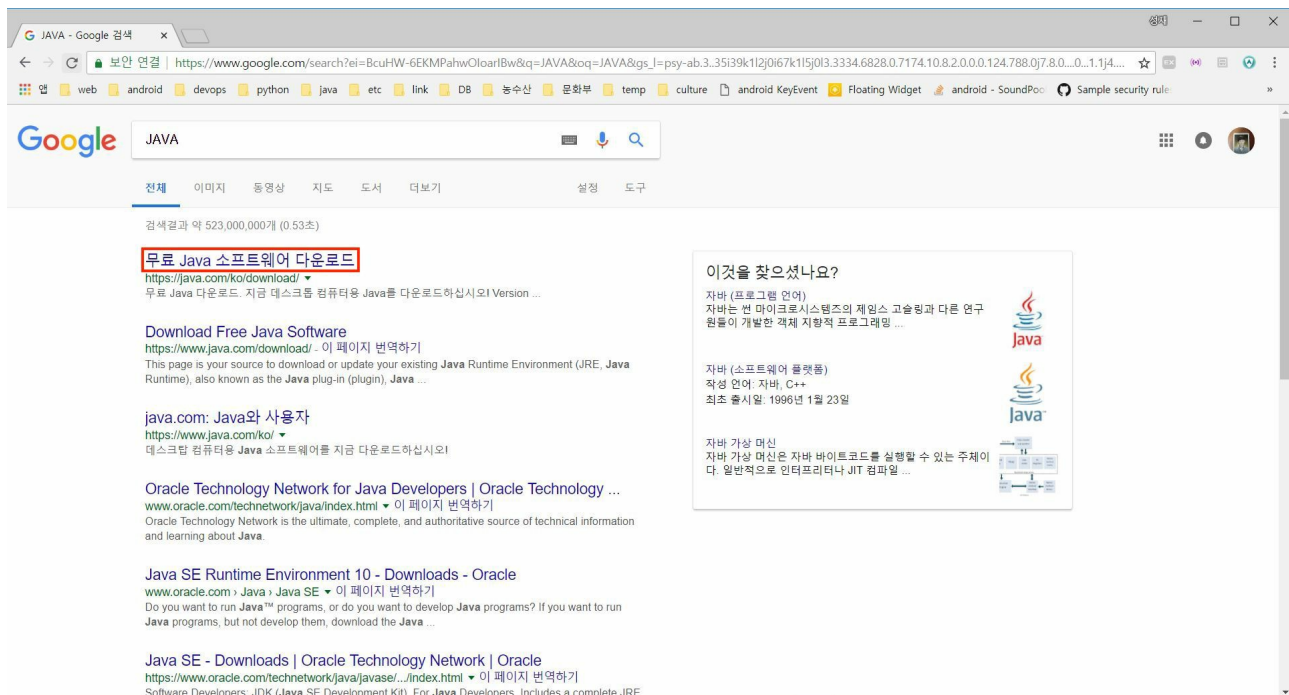
키워드 : [pip \(https://ko.wikipedia.org/wiki/Pip\\_\(패키지\\_관리자\)\)](https://ko.wikipedia.org/wiki/Pip_(패키지_관리자)),  
[requests \(http://docs.python-requests.org/en/master/\)](http://docs.python-requests.org/en/master/),  
[bs4 \(https://www.crummy.com/software/BeautifulSoup/bs4/doc/\)](https://www.crummy.com/software/BeautifulSoup/bs4/doc/),  
[openpyxl \(https://openpyxl.readthedocs.io/en/stable/\)](https://openpyxl.readthedocs.io/en/stable/),  
[konlpy \(http://konlpy.org/en/latest/\)](http://konlpy.org/en/latest/)

아직 파이썬 설치와 가상 환경 설정이 이루어지지 않으신 분은 파이썬 소개 참고 자료를 보며 파이썬 설치 및 가상 환경 설정을 먼저 진행해주세요! 아울러, 강의에서 다루지 않는 Python 기본 문법 부분도 확인 부탁드립니다!

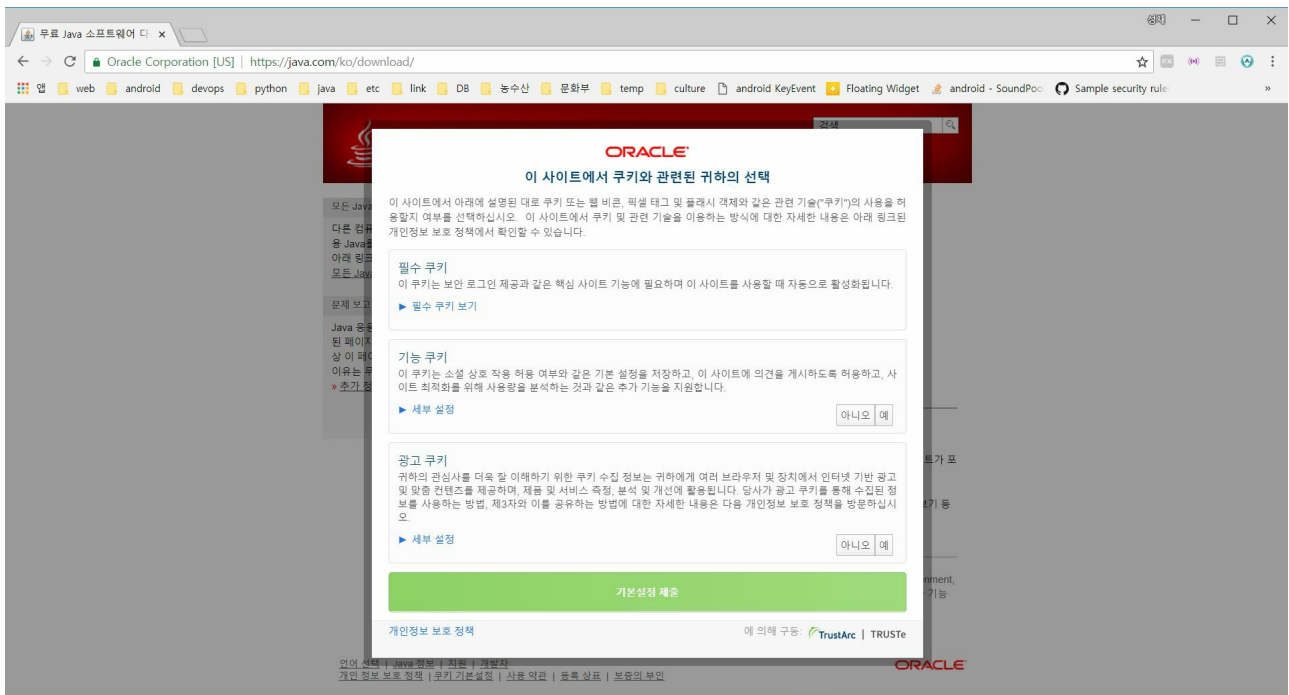
## Windows 운영체제

윈도우 운영체제를 사용하시는 분은 아래 패키지를 설치하기 전에 java와 JPyce를 설치해주셔야 합니다. 맥 운영체제를 사용하시는 분은 이 부분은 건너뛰고 바로 패키지 설치를 진행하시면 됩니다.

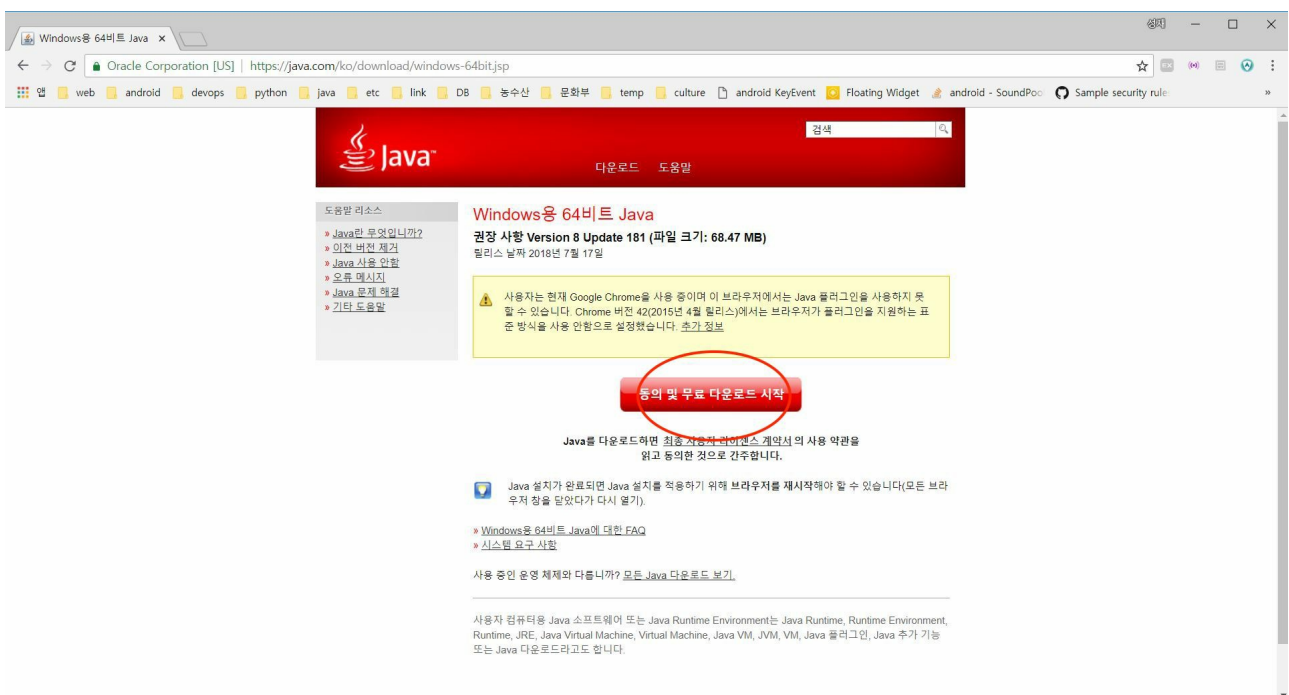
### 자바 설치



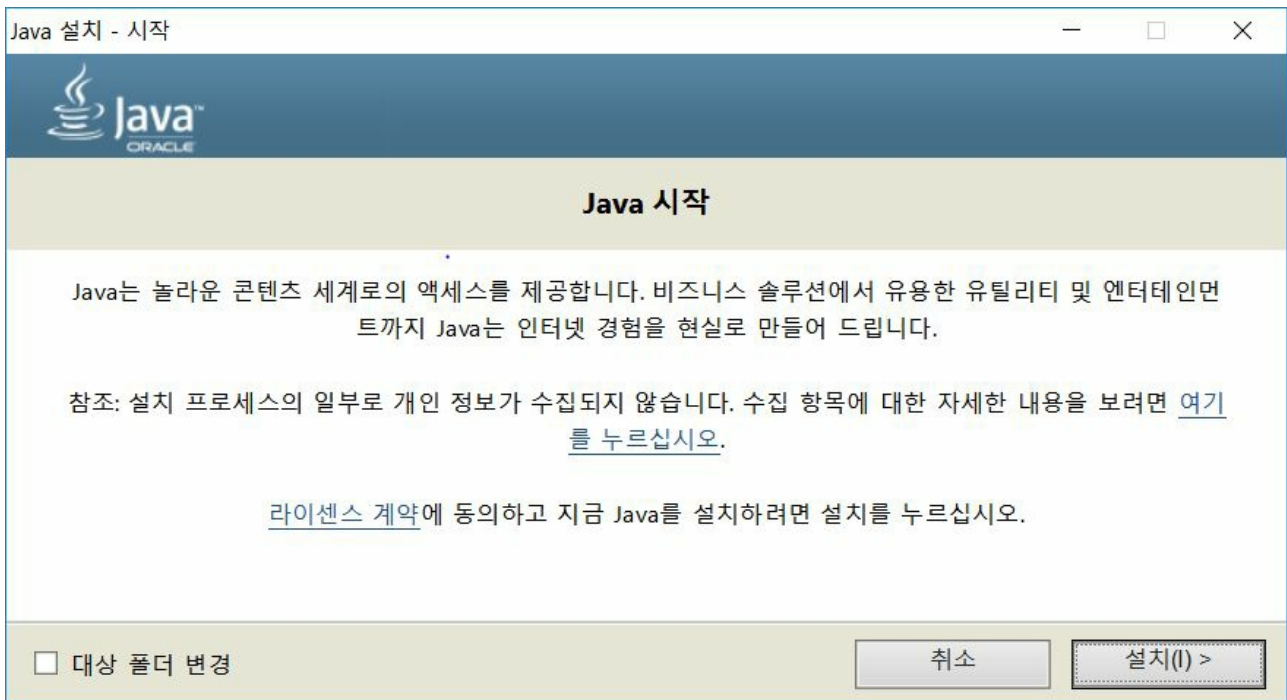
구글 검색 창에 java를 입력합니다. 가장 상단에 나오는 "무료 소프트웨어 다운로드"를 선택합니다.



경우에 따라서, 쿠키 설정 동의 요청 창이 나올 수가 있습니다. 모두 예를 클릭하고, 기본 설정 제출을 클릭합니다.



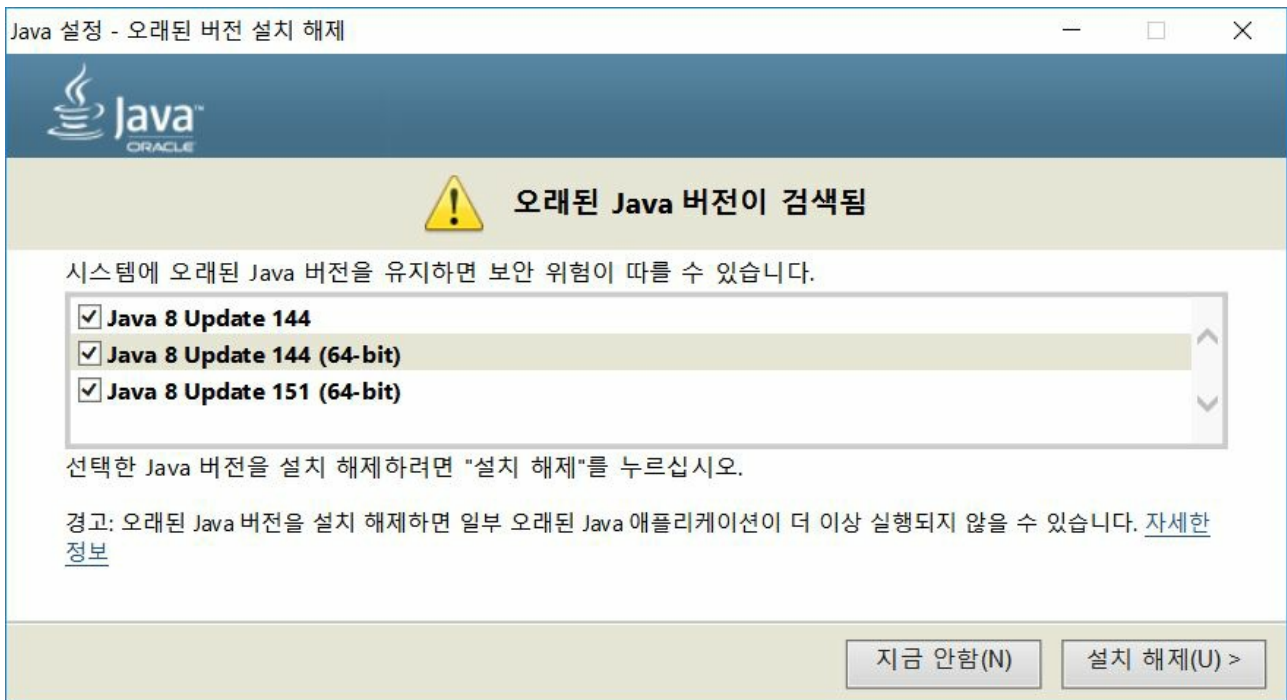
동의 및 무료 다운로드 시작을 클릭해줍니다.



설치 마법사가 나오면 설치를 클릭하여 설치를 시작합니다.



잠시 기다리면, 설치가 진행됩니다.

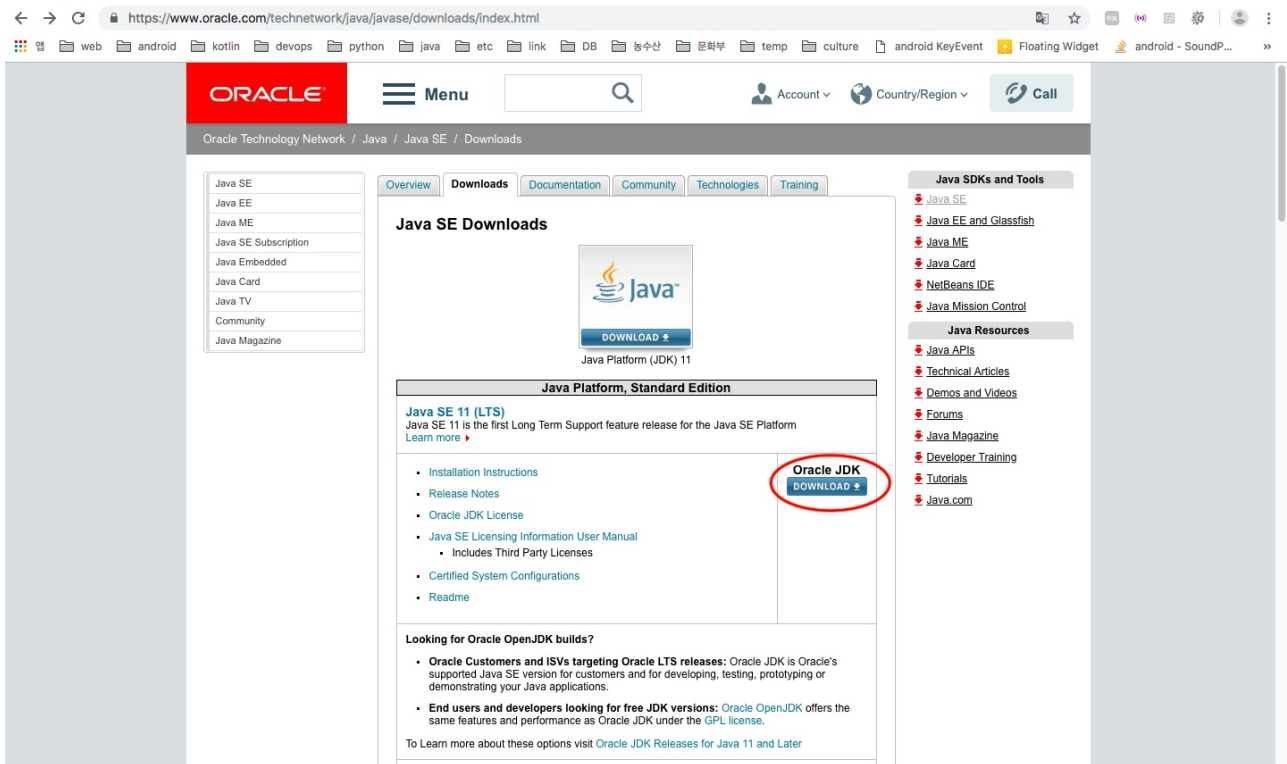


만약, 기존의 구 버전 자바가 설치되었다면, 위와 같은 화면이 나오게 됩니다. 설치 해제를 클릭해줍니다. (일정 시간 이상 소요될 수 있습니다.)

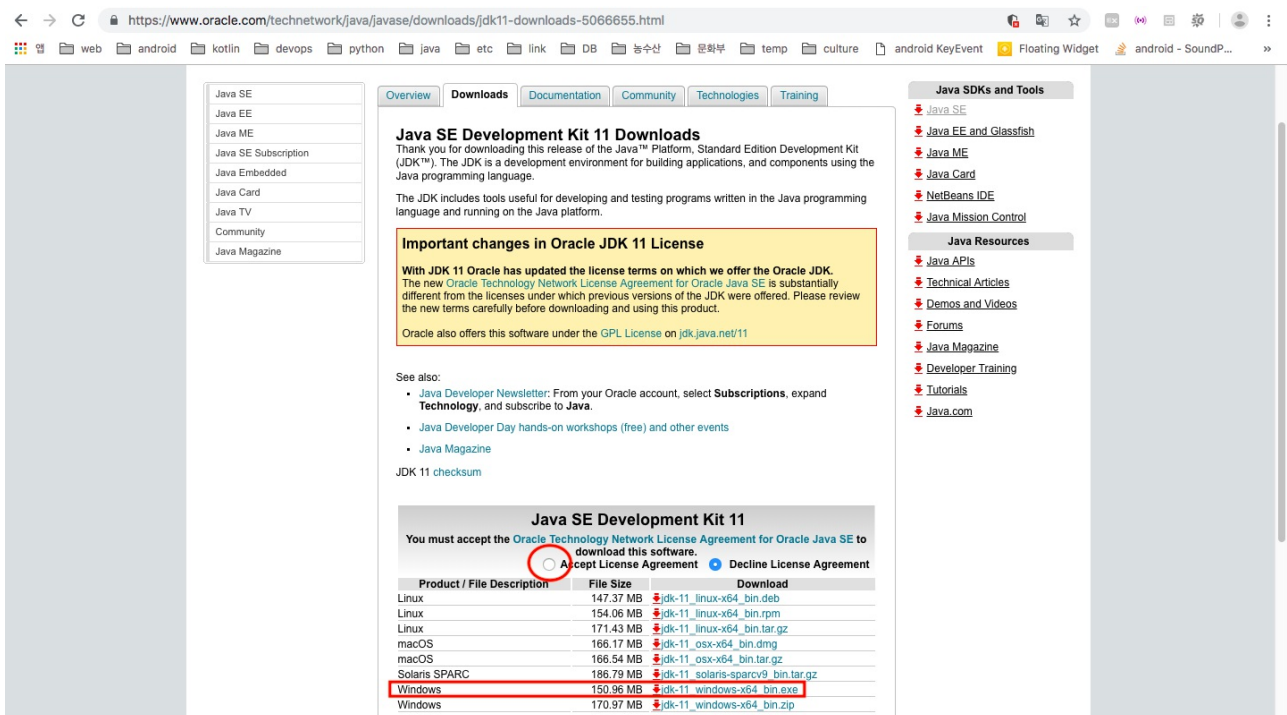


잠시 기다리면 자바 설치가 완료됩니다.

## JDK 설치



다음으로, JDK를 설치해야 합니다. [여기](https://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html)  
<https://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>)에  
 접속합니다. Oracle JDK부분을 클릭합니다.

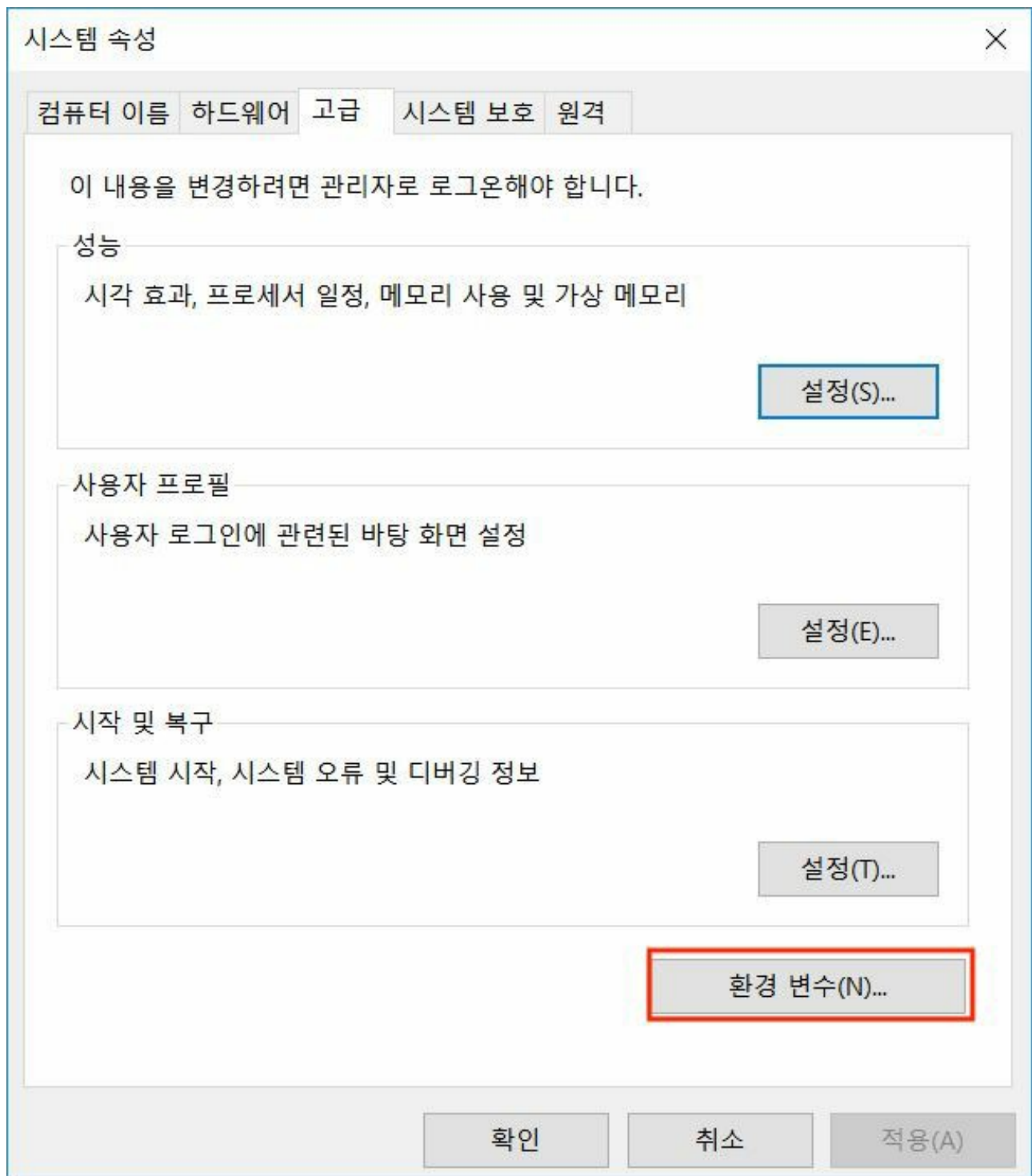


페이지 하단에 Accept License Agreement를 체크하고, windows-x64\_bin.exe 파일을 다운로드 받고  
 설치합니다.

## JAVA\_HOME 설정

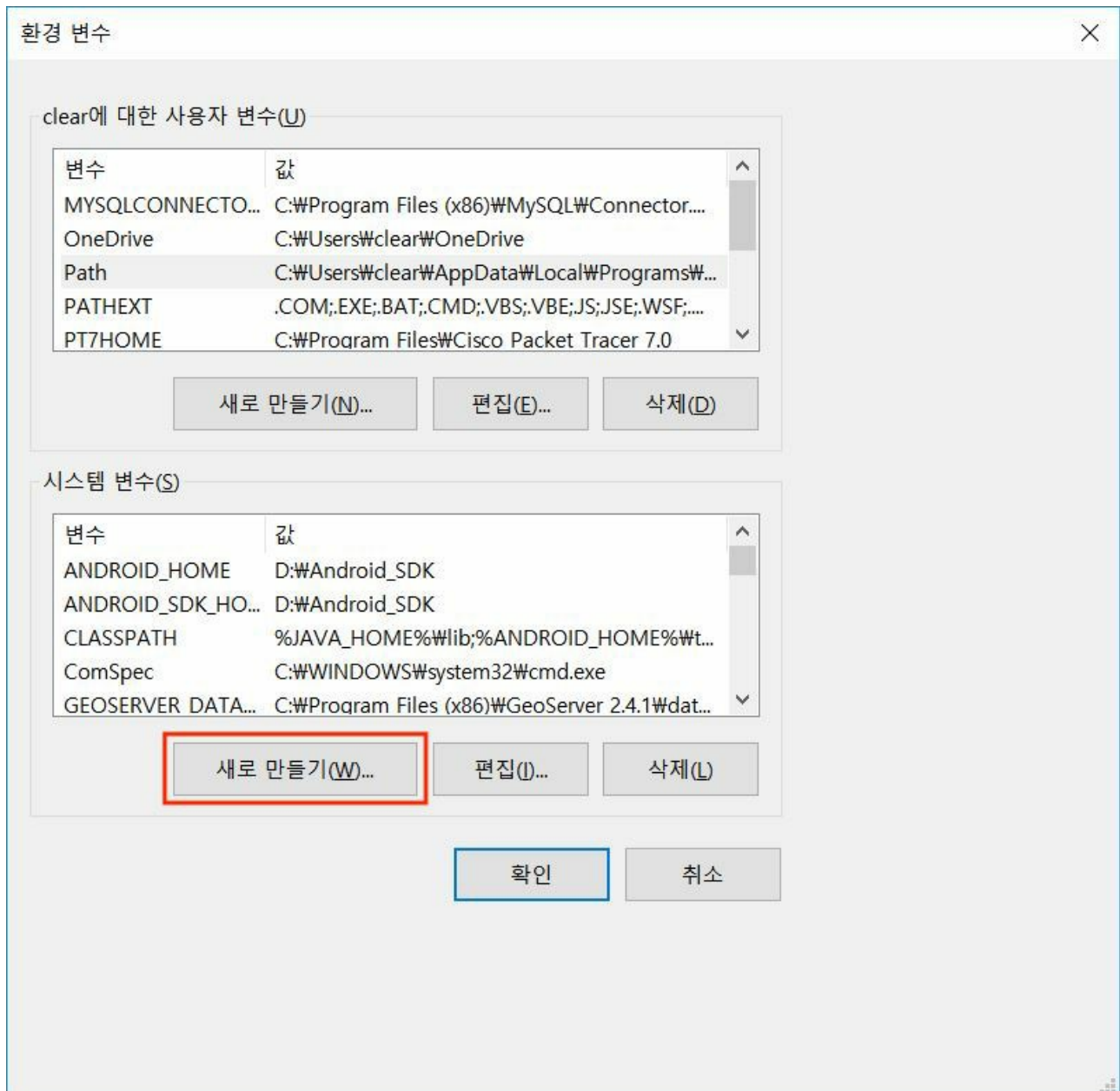
자바 설치가 완료되면, JAVA\_HOME 설정을 진행해야 합니다. 우선, java 설치 파일의 경로를 확인합니다.

보통, 자바를 처음 설치하면 자동으로 C 드라이브에 ProgramFiles 폴더 밑에 java 파일이 설치됩니다. 해당 경로에 접속하면, "jdk\_버전", "jre\_버전" 형태로 폴더가 존재합니다. jdk\_버전 폴더에 접속하여 bin 폴더를 찾습니다. 해당 폴더의 경로를 클립보드에 복사합니다. (e.g., C:\Program Files\Java\jdk1.8.0\_181\bin) 다음으로, 윈도우 탐색기에 시스템 환경 변수를 입력하면 환경 변수를 설정할 수 있는 선택 메뉴가 나타납니다. 클릭해줍니다.

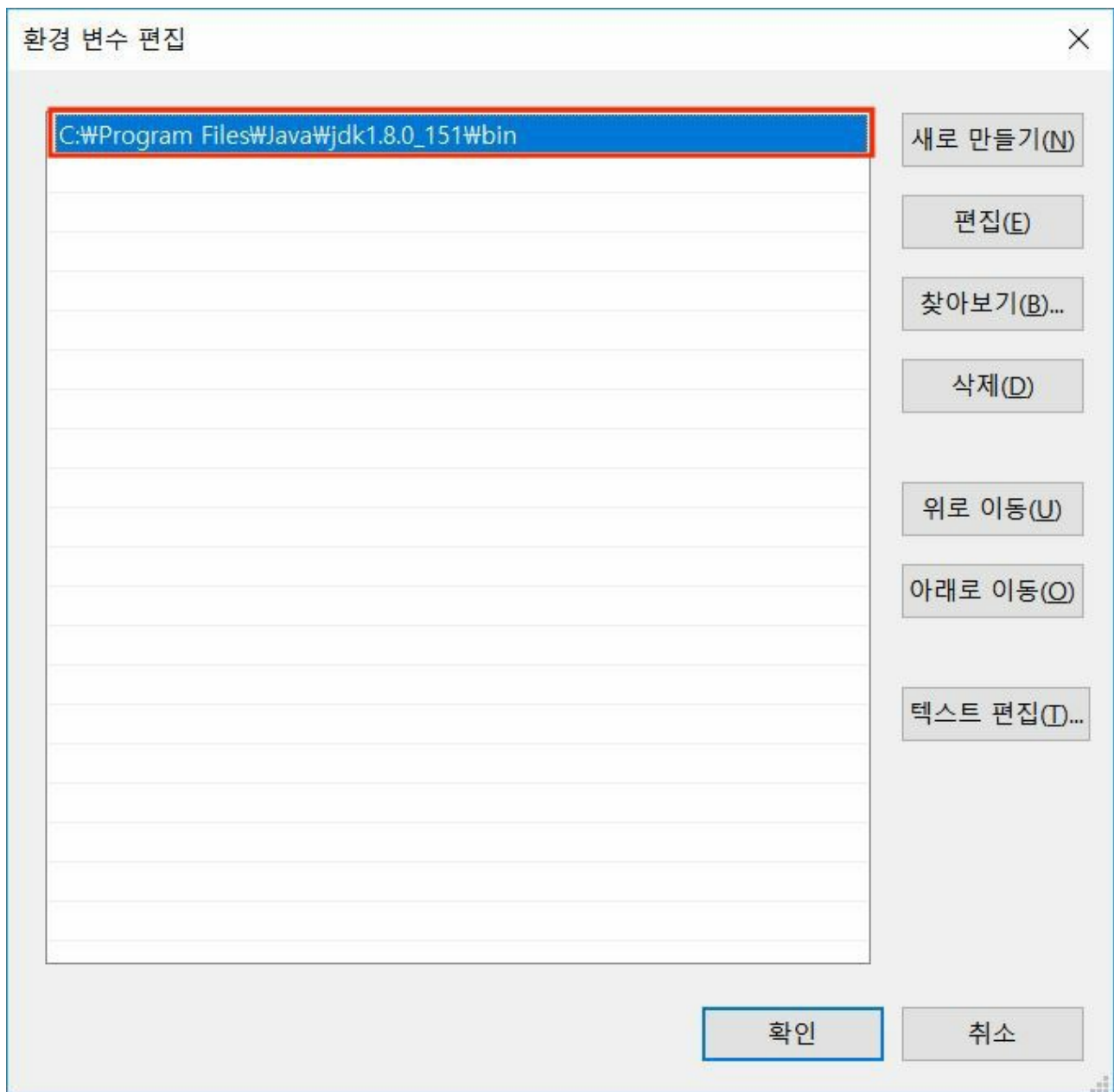


시스템 속성 창이 나타나면, "환경 변수"를 클릭해줍니다.





하단의 시스템 변수 설정 부분에서 "새로 만들기"를 클릭합니다.



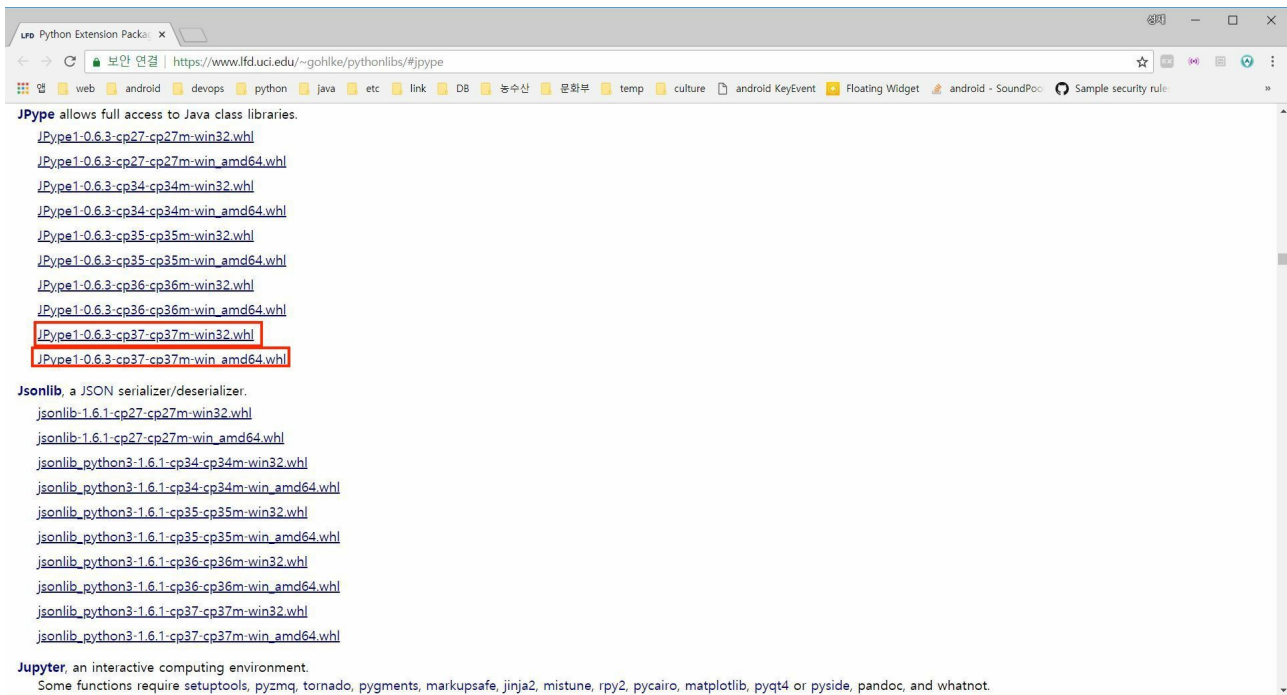
새 시스템 변수 창이 나타나면, 변수 이름을 JAVA\_HOME이라고 작성하고, 변수 값을 앞서 복사한 java/jdk버전/bin 폴더 경로를 붙여 넣기 합니다. 입력이 완료되면, 모두 확인을 누르고 빠져나옵니다.

- 윈도우 10 이하 버전(7 or 8)을 사용하시는 분은 JAVA\_HOME 변수 부분에 java/jdk버전 까지만 입력하고 확인을 누릅니다. 다음으로, 시스템 변수의 변수 이름이 Path인 것을 찾습니다. Path의 변수 값의 마지막 부분에 ;%JAVA\_HOME%\bin;을 추가합니다. (처음과 끝에 세미콜론(;))을 붙여줍니다.) 입력이 완료되면, 모두 확인을 누르고 빠져나옵니다.

## Jpype 설치

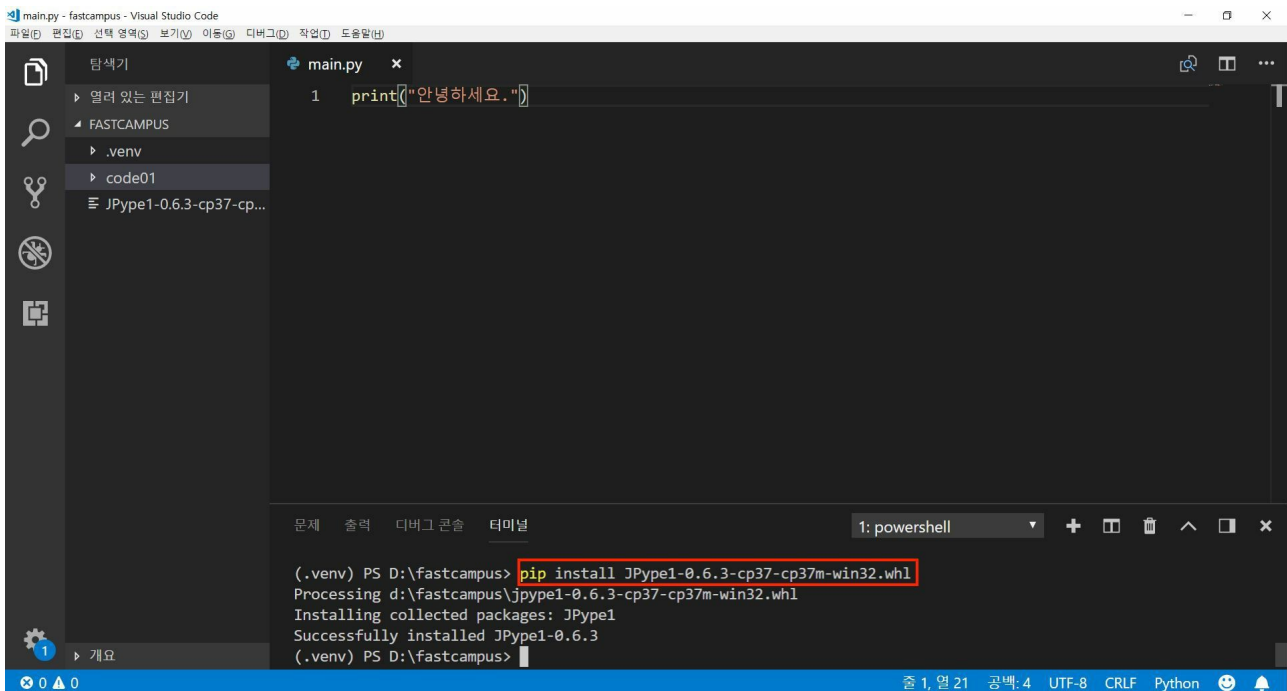
우선, [여기 \(https://www.lfd.uci.edu/~gohlke/pythonlibs/#jpype\)](https://www.lfd.uci.edu/~gohlke/pythonlibs/#jpype)에서 Jpype를 로컬 저장소에 다운로드하여야 합니다.





여러 가지 버전이 존재하는데, 그중에서 32비트 파이썬을 설치한 경우에는, JPype1-0.6.3-cp37-cp37m-win32.whl를 선택하고 다운로드해줍니다. 64비트 파이썬을 설치한 경우에는, JPype1-0.6.3-cp37-cp37m-win\_amd64.whl를 선택하고 다운로드해줍니다.

저장 경로는 따로 특정 지을 필요는 없지만, 원활한 진행을 위해서 파이썬 개발을 위해 만들어진 프로젝트 폴더를 지정하여 저장해줍니다.



VS Code 실행하고, 가상 환경을 실행합니다. (. .venv/Scripts/activate) 가상 환경이 실행되면, 아래 명령어를 입력합니다. 잠시 기다리면, 설치가 완료됩니다!

- 32비트 파이썬의 경우

```
pip install JPype1-0.6.3-cp37-cp37m-win32.whl
```

- 64비트 파이썬의 경우

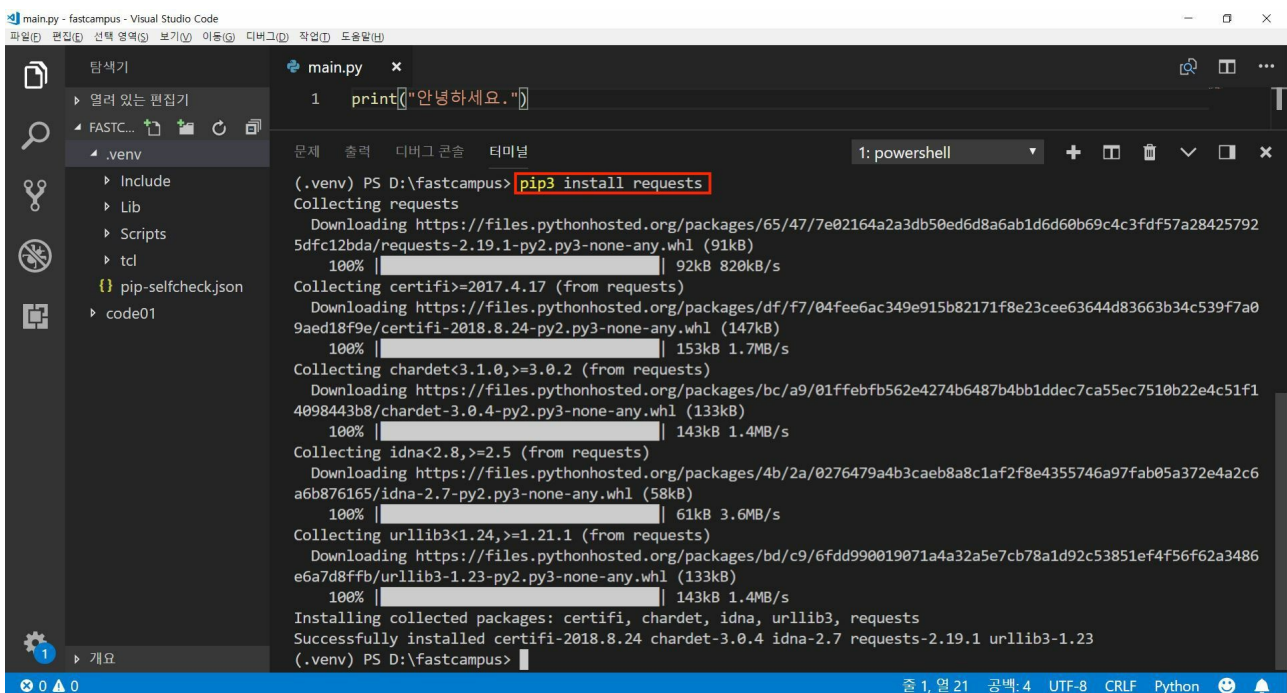
```
pip install JPyPe1-0.6.3-cp37-cp37m-win_amd64.whl
```

## 패키지 설치

우리 강의에 최종 목표인 파이썬 크롤러 개발을 위해서, 우리는 기존에 잘 만들어진 여러 패키지를 활용합니다. 같은 기능을 할 수 있는 많은 패키지가 있지만, 원활한 진행을 위해 아래 패키지를 설치해주세요! (가상 환경이 실행되어 있지 않다면, 패키지를 설치하기 전에, 가상 환경을 실행해주세요.)

### requests

requests 패키지는 웹 자원 요청을 위한 다양한 기능을 제공하는 패키지입니다.



```
main.py - fastcampus - Visual Studio Code
main.py x
1 print("안녕하세요.")

문제 출력 디버그 콘솔 터미널
1: powershell

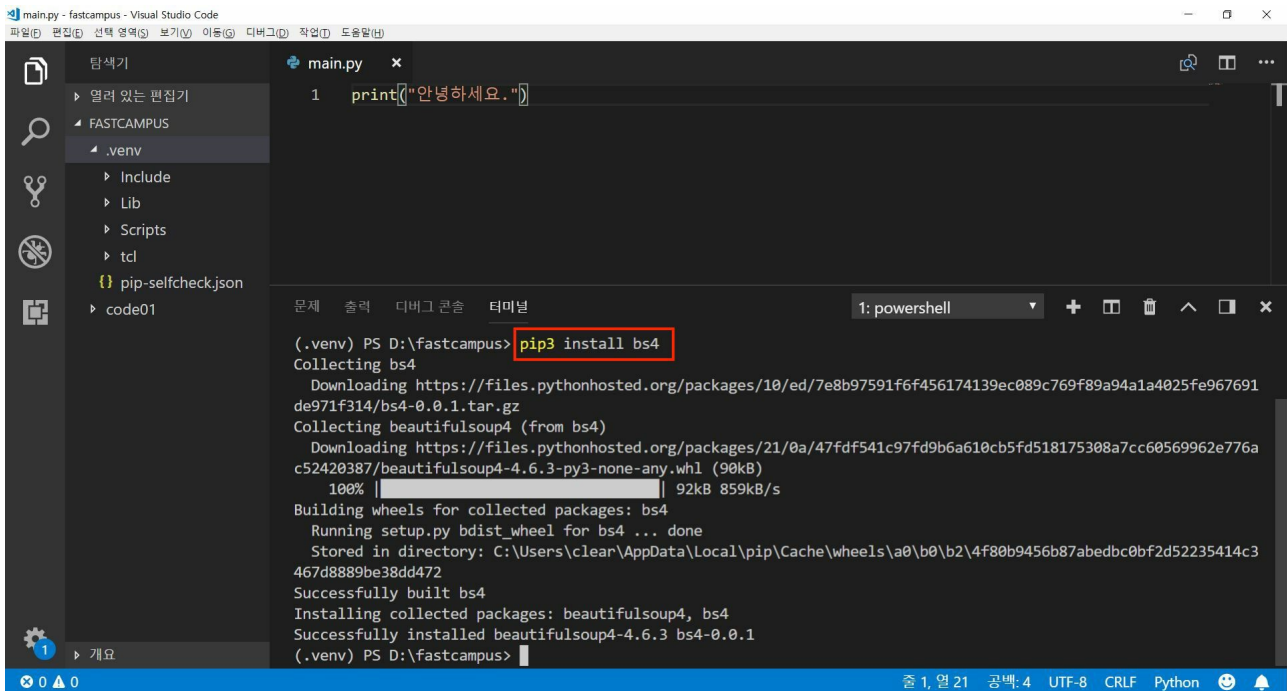
(.venv) PS D:\fastcampus> pip3 install requests
Collecting requests
  Downloading https://files.pythonhosted.org/packages/65/47/7e02164a2a3db50ed6d8a6ab1d6d60b69c4c3fdf57a284257925dfc12bda/requests-2.19.1-py2.py3-none-any.whl (91kB)
    100% |#####| 92kB 820kB/s
Collecting certifi>=2017.4.17 (from requests)
  Downloading https://files.pythonhosted.org/packages/df/f7/04fee6ac349e915b82171f8e23cee63644d83663b34c539f7a09aed18f9e/certifi-2018.8.24-py2.py3-none-any.whl (147kB)
    100% |#####| 153kB 1.7MB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests)
  Downloading https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1ddec7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl (133kB)
    100% |#####| 143kB 1.4MB/s
Collecting idna<2.8,>=2.5 (from requests)
  Downloading https://files.pythonhosted.org/packages/4b/2a/0276479a4b3caeb8a8c1af2f8e4355746a97fab05a372e4a2c6a6b876165/idna-2.7-py2.py3-none-any.whl (58kB)
    100% |#####| 61kB 3.6MB/s
Collecting urllib3<1.24,>=1.21.1 (from requests)
  Downloading https://files.pythonhosted.org/packages/bd/c9/6fdd990019071a4a32a5e7cb78a1d92c53851ef4f56f62a3486e6a7d8fffb/urllib3-1.23-py2.py3-none-any.whl (133kB)
    100% |#####| 143kB 1.4MB/s
Installing collected packages: certifi, chardet, idna, urllib3, requests
Successfully installed certifi-2018.8.24 chardet-3.0.4 idna-2.7 requests-2.19.1 urllib3-1.23
(.venv) PS D:\fastcampus>
```

아래 명령어를 입력해주세요.

```
pip3 install requests
```

### bs4

bs4 패키지는 다양한 형식으로 웹 자원을 파싱(Parsing)하고, 분석할 수 있게 해주는 패키지입니다.



```
main.py - fastcampus - Visual Studio Code
파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 디버그(D) 작업(T) 도움말(H)

탐색기
  열려 있는 편집기
  FASTCAMPUS
  .venv
    Include
    Lib
    Scripts
    tcl
  pip-selfcheck.json
  code01

main.py x
1 print("안녕하세요.")

문제 출력 디버그 콘솔 터미널
1: powershell

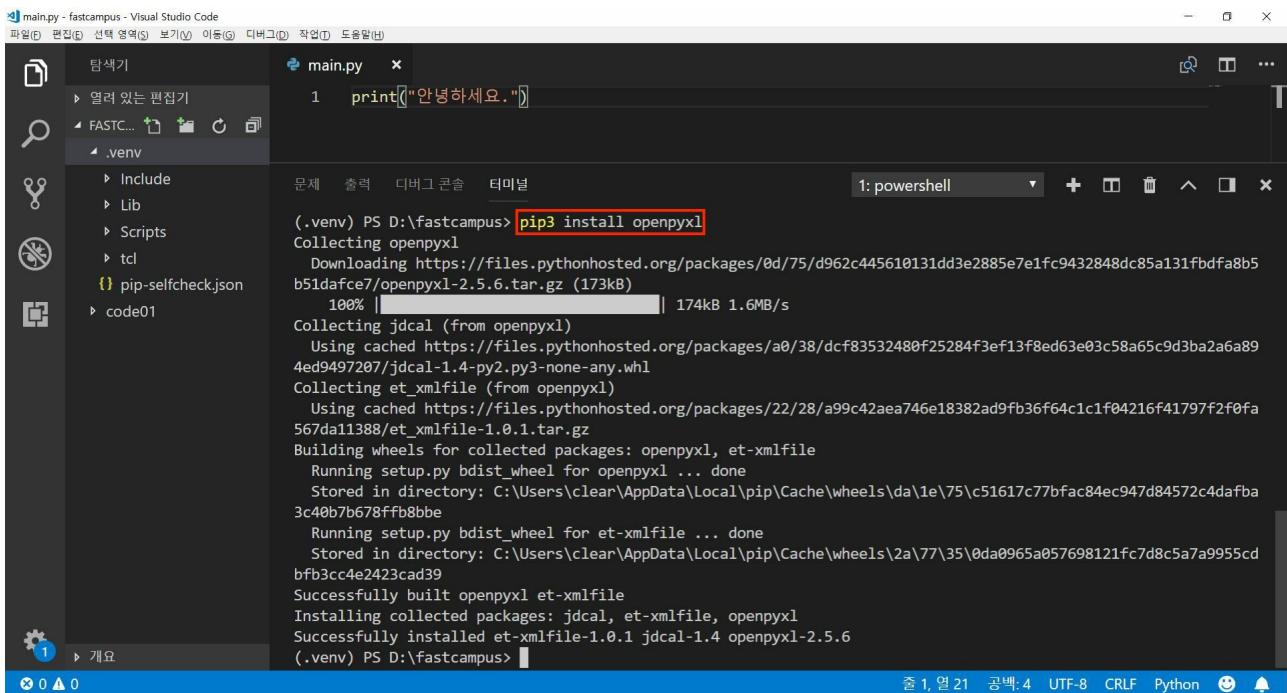
(.venv) PS D:\fastcampus> pip3 install bs4
Collecting bs4
  Downloading https://files.pythonhosted.org/packages/10/ed/7e8b97591f6f456174139ec089c769f89a94a1a4025fe967691de971f314/bs4-0.0.1.tar.gz
Collecting beautifulsoup4 (from bs4)
  Downloading https://files.pythonhosted.org/packages/21/0a/47fdf541c97fd9b6a610cb5fd518175308a7cc60569962e776ac52420387/beautifulsoup4-4.6.3-py3-none-any.whl (90kB)
100% |#####| 92kB 859kB/s
Building wheels for collected packages: bs4
  Running setup.py bdist_wheel for bs4 ... done
  Stored in directory: C:\Users\clear\AppData\Local\pip\Cache\wheels\ao\b0\b2\4f80b9456b87abedbc0bf2d52235414c3467d8889be38dd472
Successfully built bs4
Installing collected packages: beautifulsoup4, bs4
Successfully installed beautifulsoup4-4.6.3 bs4-0.0.1
(.venv) PS D:\fastcampus>
```

아래 명령어를 입력해주세요.

```
pip3 install bs4
```

## openpyxl

openpyxl 패키지는 엑셀 처리를 할 수 있는 패키지입니다.



```
main.py - fastcampus - Visual Studio Code
파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 디버그(D) 작업(T) 도움말(H)

탐색기
  열려 있는 편집기
  FASTC...
  .venv
    Include
    Lib
    Scripts
    tcl
  pip-selfcheck.json
  code01

main.py x
1 print("안녕하세요.")

문제 출력 디버그 콘솔 터미널
1: powershell

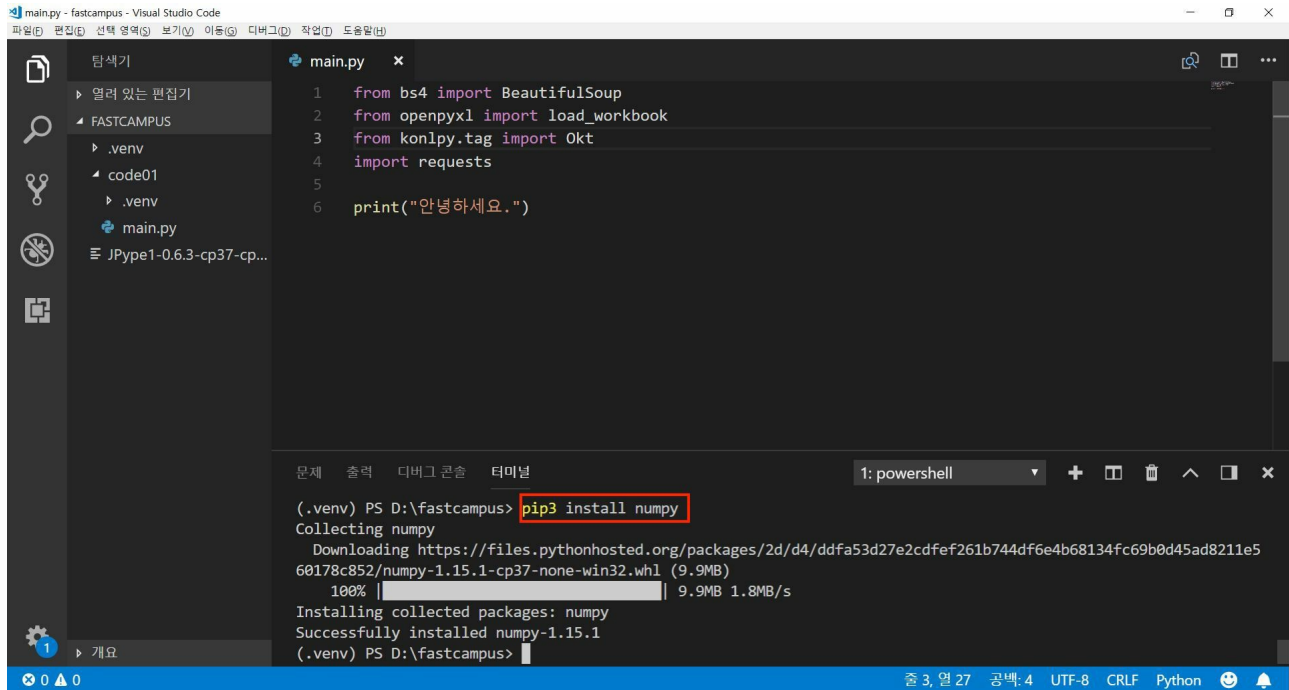
(.venv) PS D:\fastcampus> pip3 install openpyxl
Collecting openpyxl
  Downloading https://files.pythonhosted.org/packages/0d/75/d962c445610131dd3e2885e7e1fc9432848dc85a131fbdfa8b5b51dafce7/openpyxl-2.5.6.tar.gz (173kB)
100% |#####| 174kB 1.6MB/s
Collecting jdcalf (from openpyxl)
  Using cached https://files.pythonhosted.org/packages/a0/38/dcf83532480f25284f3ef13f8ed63e03c58a65c9d3ba2a6a894ed9497207/jdcalf-1.4-py2.py3-none-any.whl
Collecting et_xmlfile (from openpyxl)
  Using cached https://files.pythonhosted.org/packages/22/28/a99c42aea746e18382ad9fb36f64c1c1f04216f41797f2f0fa567da11388/et_xmlfile-1.0.1.tar.gz
Building wheels for collected packages: openpyxl, et_xmlfile
  Running setup.py bdist_wheel for openpyxl ... done
  Stored in directory: C:\Users\clear\AppData\Local\pip\Cache\wheels\da\1e\75\c51617c77bfac84ec947d84572c4dafba3c40b7b678ffb8bbe
  Running setup.py bdist_wheel for et_xmlfile ... done
  Stored in directory: C:\Users\clear\AppData\Local\pip\Cache\wheels\2a\77\35\0da0965a057698121fc7d8c5a7a9955cd bfb3cc4e2423cad39
Successfully built openpyxl et_xmlfile
Installing collected packages: jdcalf, et_xmlfile, openpyxl
Successfully installed et_xmlfile-1.0.1 jdcalf-1.4 openpyxl-2.5.6
(.venv) PS D:\fastcampus>
```

아래 명령어를 입력해주세요.

```
pip3 install openpyxl
```

## numpy

numpy 패키지는 다양한 수학 관련 처리를 할 수 있는 패키지입니다. 강의에서 사용되는 패키지는 아니지만, konlpy 패키지 사용을 위해 필요합니다. (파이썬에서 자주 사용되는 패키지이므로, 설치해두면 두루두루 사용하는 경우가 생길 겁니다! :)



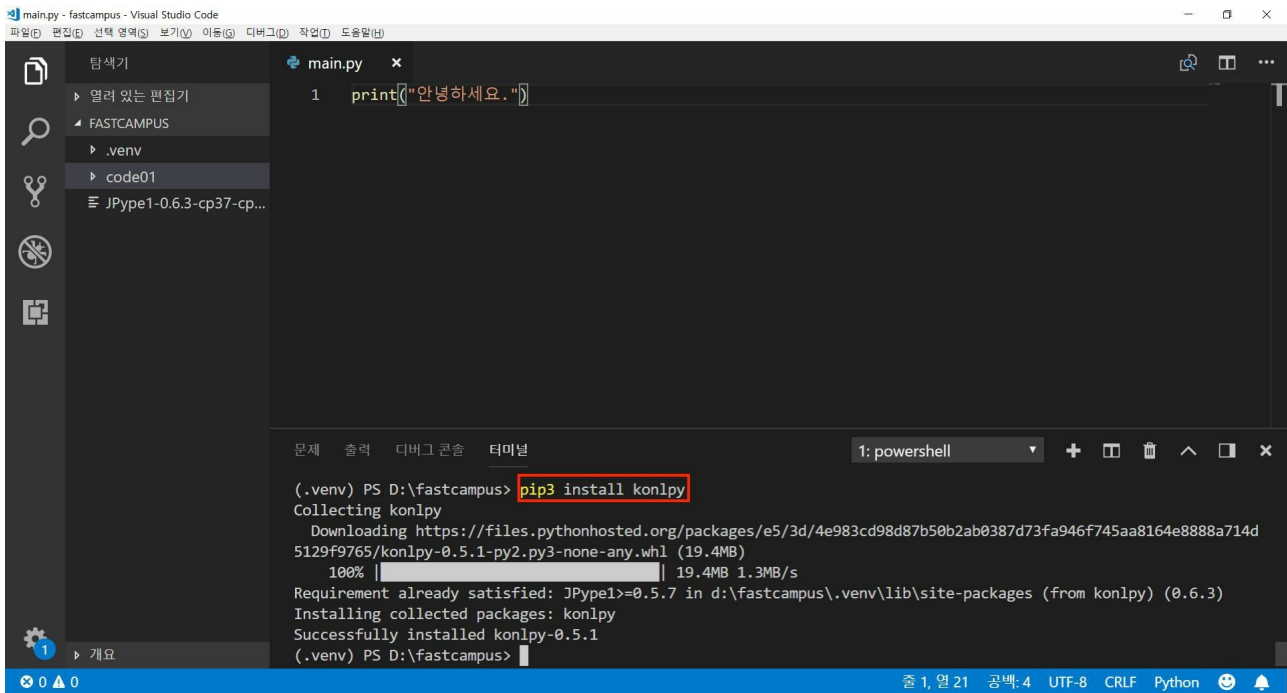
The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project structure with folders like .venv and code01, and a file named main.py. The main editor window displays the content of main.py, which contains Python code for importing BeautifulSoup, openpyxl, konlpy.tag, and requests, and printing a greeting. At the bottom, a terminal window is open, showing the command `pip3 install numpy` being executed in a PowerShell environment. The terminal output shows the progress of downloading and installing numpy-1.15.1, with a progress bar at 100% and a message 'Successfully installed numpy-1.15.1'.

아래 명령어를 입력해주세요.

```
pip3 install numpy
```

## konlpy

konlpy 패키지는 한글 분석 및 처리를 할 수 있는 패키지입니다. 위의 numpy 패키지가 설치되지 않았다면, 에러가 발생할 수 있습니다. 설치 전에 numpy 패키지를 먼저 설치해주세요!



```
main.py x
1 print("안녕하세요.")

(.venv) PS D:\fastcampus> pip3 install konlpy
Collecting konlpy
  Downloading https://files.pythonhosted.org/packages/e5/3d/4e983cd98d87b50b2ab0387d73fa946f745aa8164e8888a714d5129f9765/konlpy-0.5.1-py2.py3-none-any.whl (19.4MB)
    100% |#####| 19.4MB 1.3MB/s
Requirement already satisfied: JPype1>=0.5.7 in d:\fastcampus\.venv\lib\site-packages (from konlpy) (0.6.3)
Installing collected packages: konlpy
Successfully installed konlpy-0.5.1
(.venv) PS D:\fastcampus>
```

아래 명령어를 입력해주세요.

```
pip3 install konlpy
```

이제 정상적으로 파이썬 코드를 실행할 준비가 되었는지 확인하기 위해서, 아래 코드를 입력합니다. 파이썬 파일을 아직 만들지 않으셨다면, main.py 파일을 만들어줍니다.

```
from bs4 import BeautifulSoup
from opnepyxl import load_workbook
from konlpy.tag import Okt
import requests

print("안녕하세요.")
```

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a file tree with 'FASTCAMPUS' and 'code01' folders. The main editor displays a file named 'main.py' with the following Python code:

```
1 from bs4 import BeautifulSoup
2 from openpyxl import load_workbook
3 from konlpy.tag import Okt
4 import requests
5
6 print("안녕하세요.")
```

At the bottom, the integrated terminal shows the execution of the script:

```
(.venv) PS D:\fastcampus> cd code01
(.venv) PS D:\fastcampus\code01> python main.py
안녕하세요.
(.venv) PS D:\fastcampus\code01>
```

The status bar at the bottom indicates the file is 6 lines long, 21 characters, using UTF-8 encoding, CRLF line endings, and is a Python file.

위 코드를 입력하고, 파일이 있는 경로에 들어갑니다. (cd 경로) 터미널에 `python main.py` 명령어를 입력합니다. 터미널에 에러 없이 "안녕하세요."가 잘 나오면 성공입니다!

지금은 위 코드가 어떤 의미인지 몰라도 괜찮습니다. 모듈과 패키지 강의 부분에서 자세하게 다루게 될 내용입니다. :) 지금은, 아무런 에러 없이 외부 패키지를 잘 불러왔고, 이를 활용해서 코드를 작성할 준비가 되었다는 점만 기억해주시면 됩니다!