

codingOn x posco

K-Digital Training 신재생에너지 활용 IoT 과정

가상환경과 웹크롤링

Anaconda

Python vs Anaconda

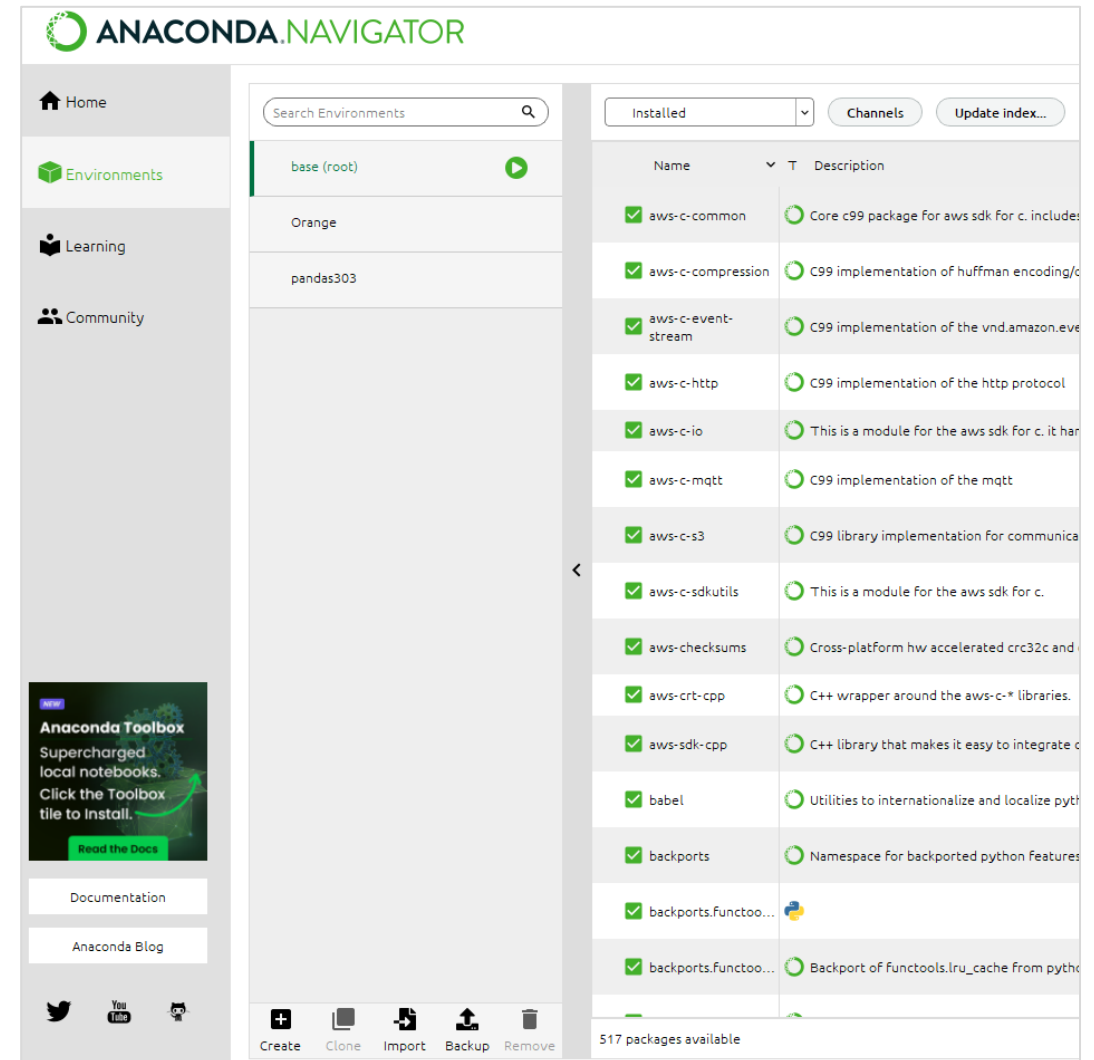
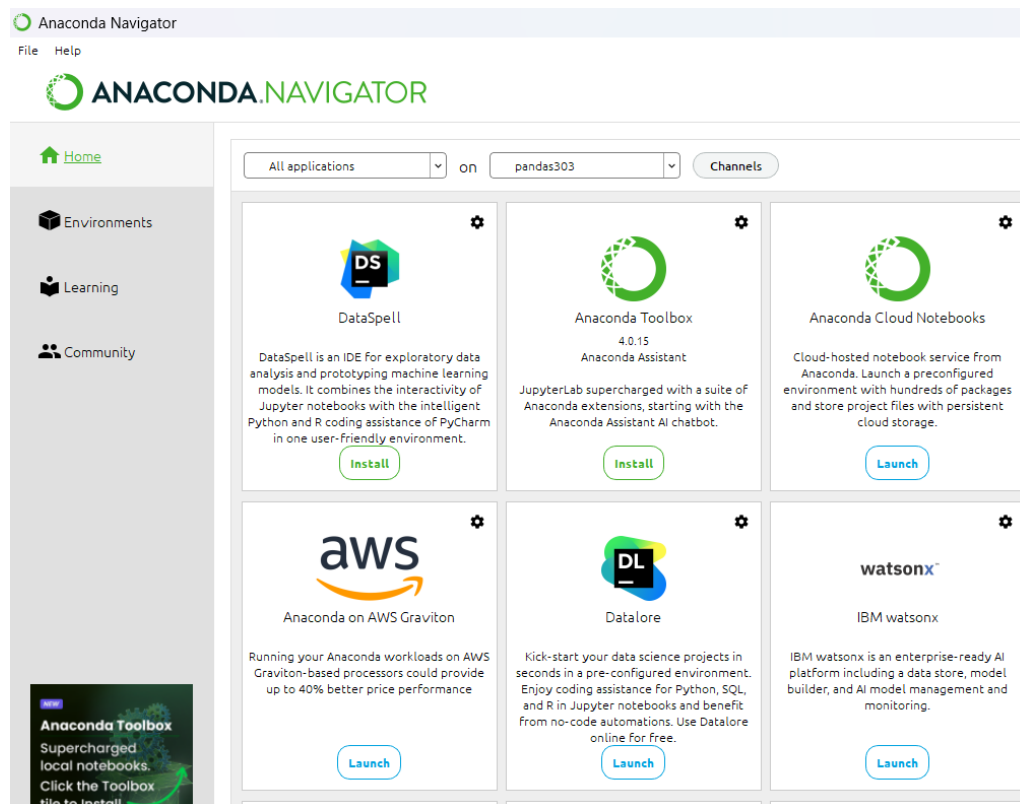


- **Python**은 기본적으로 패키지 관리 시스템인 pip만을 포함하고 있음
 - ➡ 필요한 툴, 패키지가 있다면 pip을 통해 수동으로 추가해야 함.
 - ➡ 패키지가 컴퓨터 자체에 설치됨
 - ➡ 프로젝트를 여러 번 진행하다 보면 필요한 패키지는 2~3개 정도면 되는데, 10개 15개의 패키지들이 설치되어 필요 이상으로 공간을 차지하기도 함.
- **Anaconda**는 데이터 분석, 머신 러닝 등에 사용하는 여러가지 패키지가 기본적으로 포함되어 있음

아나콘다 네비게이터 실행

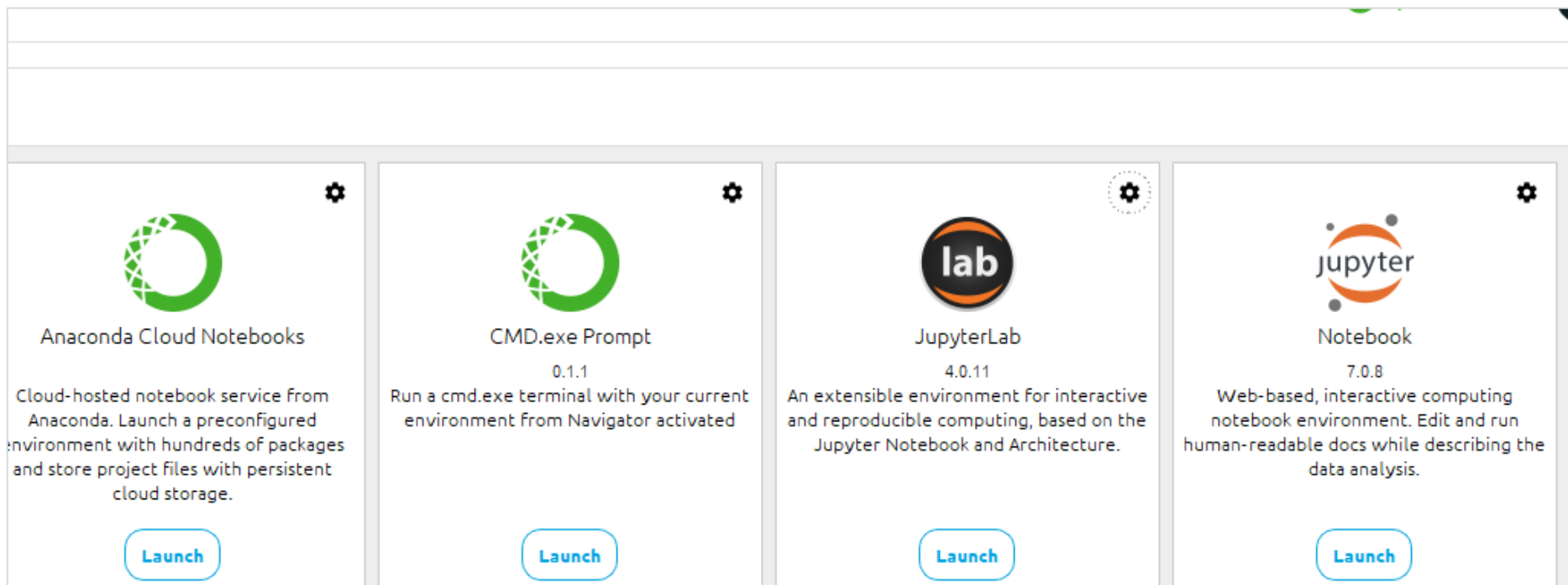
■ 아나콘다 라이브러리 설치 확인

Environments > base(root)



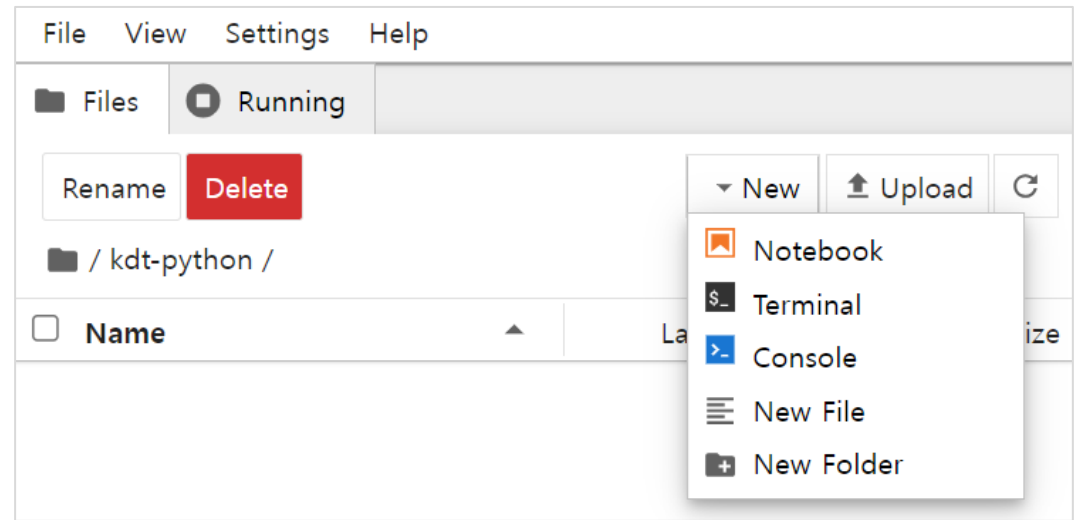
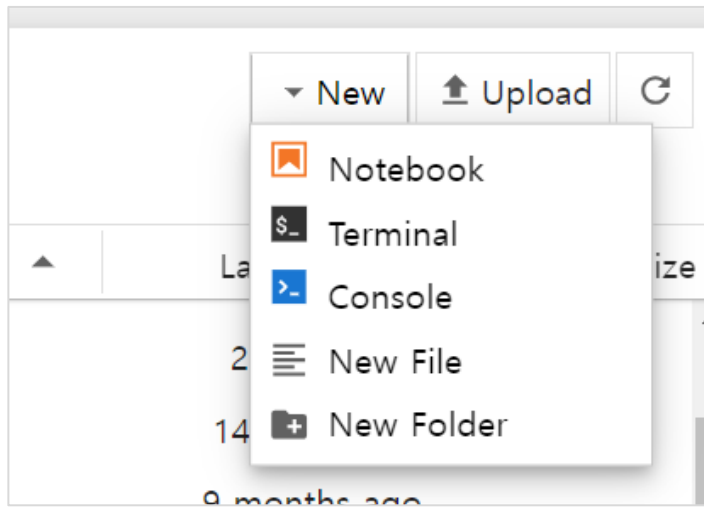
주피터 노트북 사용

■ Jupyter Notebook > Launch



주피터 노트북 사용

- New > New Folder > 신규폴더 생성



- 신규폴더 > New > Notebook

주피터 노트북 사용

- 상단 > Untitled > basic.ipynb(파일)

```
# 변수  
msg = "행운을 빌어요!!"  
# print(msg)  
msg  
  
'행운을 빌어요!!'
```

- 단축키
 - 실행: ctrl + Enter
 - 실행후 다음 줄 생성: shift + Enter

주피터 노트북 사용

```
# 리스트  
cart = ['라면', '콩나물', '계란', '초코파이']  
cart  
cart[0]  
cart[-1]  
for i in cart:  
    print(i)
```

['라면', '콩나물', '계란', '초코파이']

라면

콩나물

계란

초코파이

주피터 노트북 사용

모듈 임포트하기

```
import math
import random

v1 = math.ceil(12.56)
print(v1)
v2 = math.floor(12.56)
print(v2)

v3 = random.random()
print(v3)
v4 = random.randint(1, 10)
print(v4)

# 주사위 10번 던지기
for i in range(10):
    dice = random.randint(1, 6)
    print(dice)
```

주피터 노트북 사용

- basic2.ipynb

함수, 클래스 사용하기

```
def myabs(x):  
    if x < 0:  
        return -x  
    else:  
        return x
```

```
print(myabs(-2))  
print(myabs(2))
```

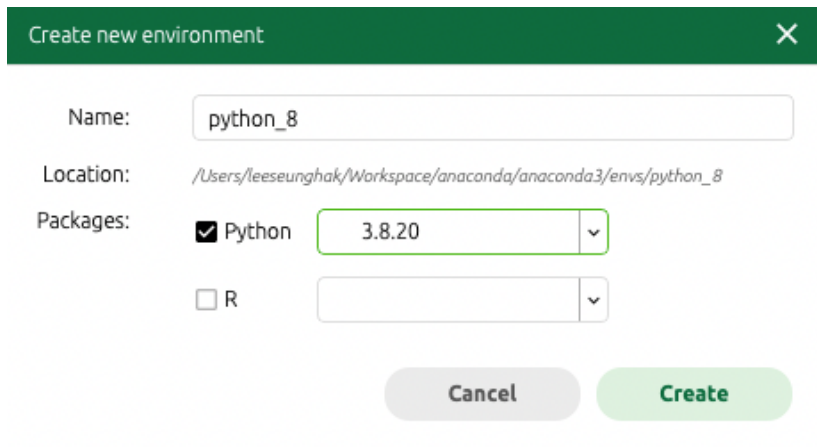
```
# 내장 abs()  
print(abs(-2))  
print(abs(2))
```

```
class Car:  
    def __init__(self, model_name, year):  
        self.model_name = model_name  
        self.year = year  
  
    def __str__(self):  
        return f'모델명: {self.model_name}, 연식: {self.year}'  
  
car1 = Car('K7', 2020)  
print(car1)  
car2 = Car('아이오닉5', 2023)  
print(car2)
```

아나콘다 가상환경

- 가상환경 생성

Environments > Create(하단 아이콘) > python_8



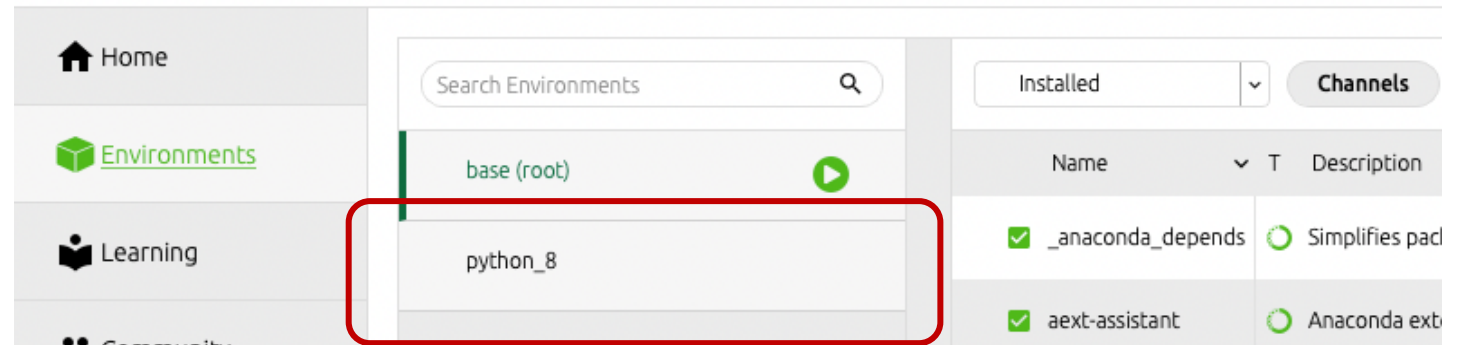
Create new environment

Name:

Location:

Packages:

- ☒ Python
- ☐ R



Home

Environments

Learning

Community

Search Environments

base (root)

python_8

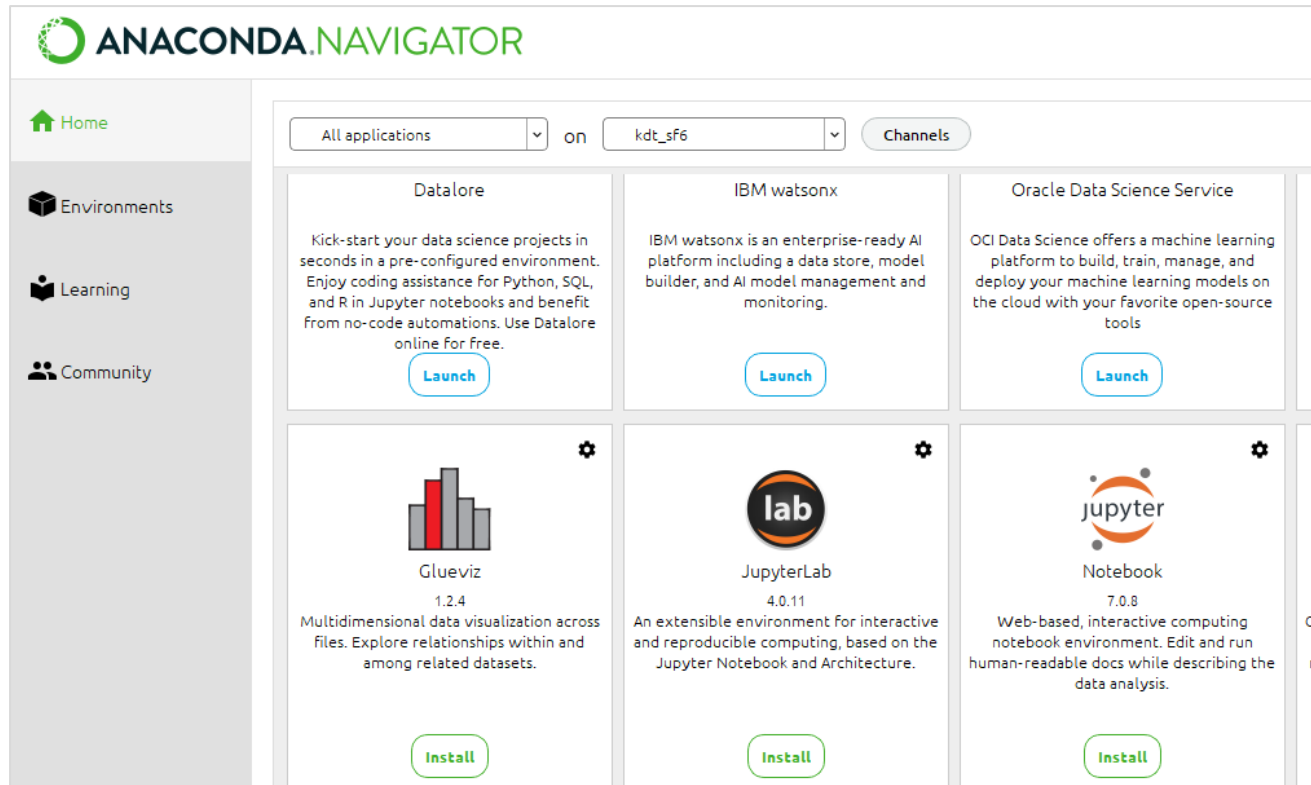
Installed

Channels

Name	T	Description
<input checked="" type="checkbox"/> _anaconda_depends		Simplifies pack
<input checked="" type="checkbox"/> aext-assistant		Anaconda ext

아나콘다 개발도구

- 개발도구(IDE) 실행
 - Home > 주피터 랩 > Install > Launch

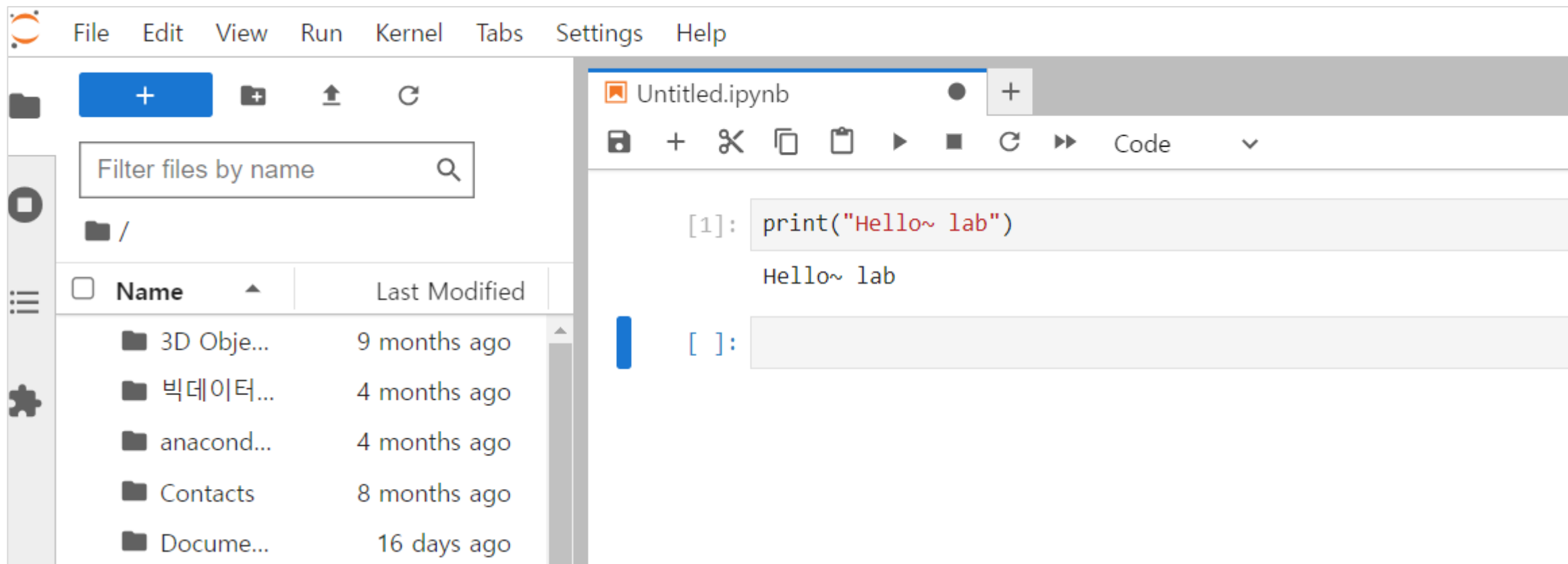


주피터 랩(Jupyter Lab)

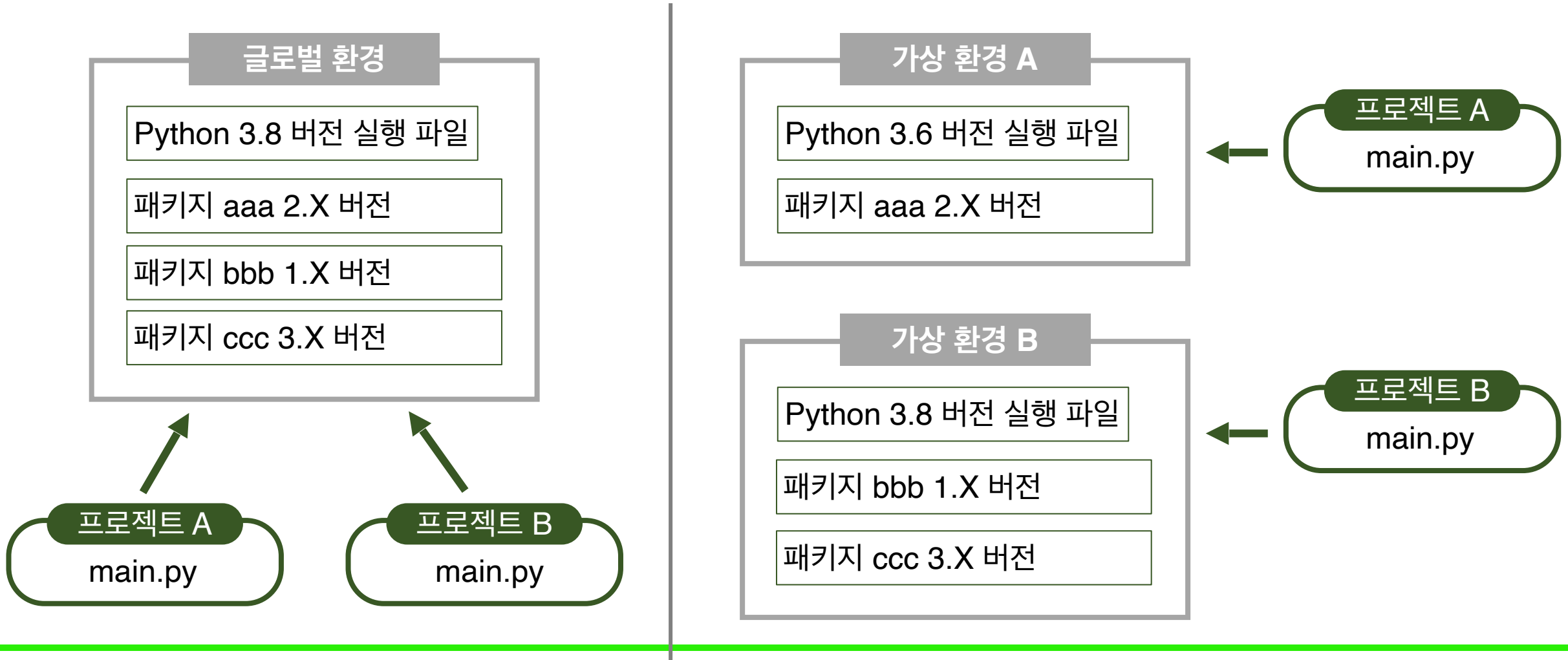
- ✓ Jupyter Lab은 2018년 출시되었고, Jupyter Notebook 보다 더 발전된 버전이다.
 - ✓ 주피터 랩은 대화형 컴퓨팅을 지원하여 코드를 실행하고 결과를 즉시 확인할 수 있다.
 - ✓ 다양한 플러그인과 확장 기능을 제공하여 사용자 정의 작업환경을 구성할 수 있다.
 - ✓ 데이터 분석 및 시각화 작업에 탁월하며 문서화 하여 저장하고 공유할 수 있다.
-

주피터 랩 실행

- Python3(ipykernel) 아이콘 클릭

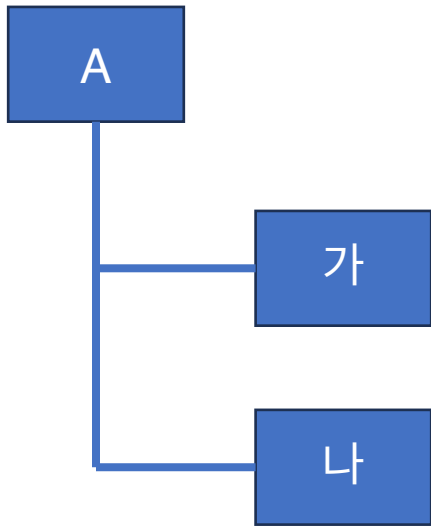


가상 환경 vs 글로벌 환경



가상 환경 vs 글로벌 환경

- 폴더구조



예) pip install 모듈

글로벌환경

- 가, 나 폴더에 모두 사용가능

가상환경

- 만약 가폴더에서만 가상환경을 생성하였다면
가폴더에서만 사용가능

가상환경이 필요한 이유?

- 프로젝트를 진행하다 보면 여러 library, package를 다운로드 하게 됨.

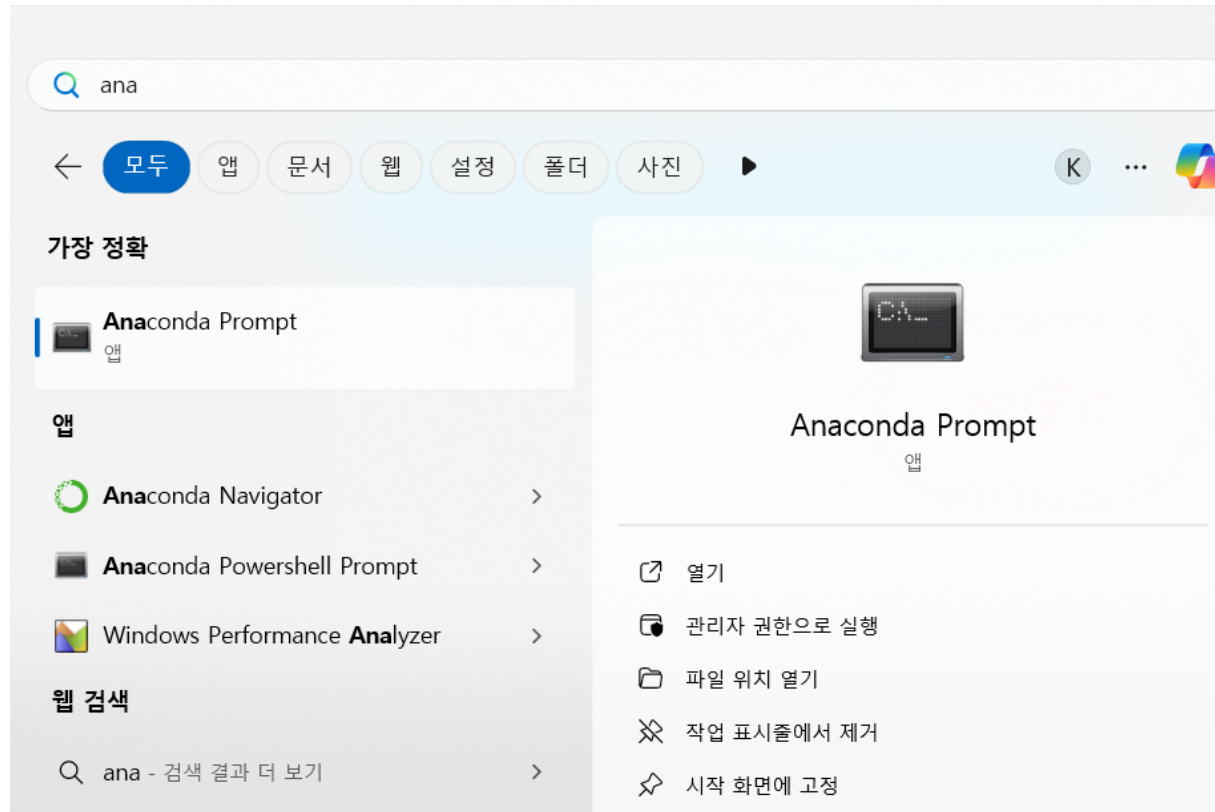
➡ 각 library, package들끼리 **충돌**을 일으키는 경우 ↑

- 이런 경우에 가상환경을 이용한다면?

➡ 프로젝트별로 **독립적인 작업 환경**에서 작업할 수 있다!

Anaconda 가상 환경 관리

- Anaconda Prompt 실행



Anaconda 가상 환경 관리

- 가상환경 조회 – conda env list

```
> conda env list

# conda environments:
#
base                /Users/leeseunghak/Workspace/anaconda/anaconda3
python_8            /Users/leeseunghak/Workspace/anaconda/anaconda3/envs/python_8
```

Anaconda 가상 환경 관리

- 새로운 가상환경 생성

가상환경 생성

> conda create -n [가상환경이름]
또는 conda create --name [가상환경이름]

파이썬 버전을 지정하며 가상환경 생성하기

> conda create -n [가상환경이름] python=[0.0]
또는 conda create --name [가상환경이름] python=[0.0]

```
> conda create -n python_9 python=3.9
```

Anaconda 가상 환경 관리

- 가상환경 활성화

1. `conda activate python_8`

2. `pip install ipykernel`

-> 주피터랩과 주피터 노트북에서 사용할 수 있는 커널 설치

- 주피터 랩 실행



```
[2]: !python --version
```

```
Python 3.8.20
```

Jupyter lab 실행

- 필수 라이브러리 설치 – 가상환경이 base(root)인 경우는 이미 설치됨
 - **!pip install requests**

```
[1]: !pip install requests
```

```
Requirement already satisfied: requests in c:\users\shg02\anaconda3\envs\kdt_test\lib\site-packages (2.32.3)  
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\shg02\anaconda3\envs\kdt_test\lib\site-packages (from requests) (3.3.2)  
Requirement already satisfied: idna<4,>=2.5 in c:\users\shg02\anaconda3\envs\kdt_test\lib\site-packages (from requests) (3.7)  
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\shg02\anaconda3\envs\kdt_test\lib\site-packages (from requests) (2.2.3)  
Requirement already satisfied: certifi>=2017.4.17 in c:\users\shg02\anaconda3\envs\kdt_test\lib\site-packages (from requests) (2024.8.30)
```

beautifulsoup

beautifulSoup

HTML과 XML 문서를 파싱하기 위한 파이썬 라이브러리이다.

웹 서버로 부터 HTML 소스코드를 가져온 다음에는 HTML 태그 구조를 해석하기 위한 과정이 필요하다.

HTML 소스 코드를 해석하는 것을 **파싱(parsing)**이라고 부른다.

▶ BeautifulSoup 설치

pip install BeautifulSoup4

▶ BeautifulSoup 사용

from bs4 import BeautifulSoup

```
soup = BeautifulSoup(html, 'html.parser')
```

beautifulSoup

```
from bs4 import BeautifulSoup

html_str = '''
<html>
  <body>
    <div id="content">
      <ul class = 'industry'>
        <li>인공지능</li>
        <li>빅데이터</li>
        <li>스마트팩토리</li>
      </ul>
      <ul class = 'comlang'>
        <li>Python</li>
        <li>C++</li>
        <li>Javascript</li>
      </ul>
    </div>
  </body>
</html>
'''

soup = BeautifulSoup(html_str, "html.parser") # html 파싱
```

beautifulsoup

- `soup.find(태그)`
 - 처음 나오는 태그로 찾기
- `soup.find_All(태그)`
 - 태그에 해당하는 모든 요소 찾아서 **리스트로 반환함**
- `soup.find(태그, attrs={'class': css_selector})`
 - 태그에 해당하는 선택자로 찾기

beautifulSoup

• find() 사용하기

```
# 처음 나오는 ul 태그로 찾기
first_ul = soup.find('ul')
print(first_ul)
print(first_ul.text) #태그 없이 텍스트 출력
```

```
# 모든 요소를 li 태그로 찾기
first_all_li = first_ul.findAll('li')
print(first_all_li)
print(first_all_li[1])
print(first_all_li[1].text)
```

```
for li in first_all_li:
    print(li.text)
```

```
# class 선택자로 찾기
# second_ul = soup.find('ul', attrs={'class': 'comlang'})
second_ul = soup.find('ul', class_='comlang')
print(second_ul)
print(second_ul.text)
```

```
<ul class="industry">
<li>인공지능</li>
<li>빅데이터</li>
<li>스마트팩토리</li>
</ul>
```

인공지능
빅데이터
스마트팩토리

```
[<li>인공지능</li>, <li>빅데이터</li>, <li>스마트팩토리</li>]
<li>빅데이터</li>
빅데이터
인공지능
빅데이터
스마트팩토리
<ul class="comlang">
<li>Python</li>
<li>C++</li>
<li>Javascript</li>
</ul>
```

beautifulsoup

- `soup.select_one(태그요소.선택자이름)`
 - `css_selector` 에 해당하는 첫번째 태그 가져옴
- `soup.select(태그요소.선택자이름)`
 - `css_selector` 에 해당하는 모든 태그 **리스트 가져옴**
- `{tag}.get_text()`
 - 해당 태그의 텍스트 가져옴

beautifulsoup

- select() 사용하기

```
# select_one(태그이름.선택자이름) - 1개 요소 찾기
first_ul = soup.select_one('ul.industry')
print(first_ul)
print(first_ul.text)  #태그 없이 텍스트 출력

# select(태그이름.선택자이름) - 모든 요소 찾기
first_all_li = first_ul.select('ul.industry > li')
print(first_all_li)
print(first_all_li[1])
print(first_all_li[1].text)

second_ul = soup.select_one('ul.comlang')
print(second_ul)
print(second_ul.text)
```

```
<ul class="comlang">
<li>Python</li>
<li>C++</li>
<li>Javascript</li>
</ul>
```

```
Python
C++
Javascript
```

데이터 크롤링시 주의사항

- 특정 웹사이트의 페이지를 쉬지 않고 크롤링하는 행위를 무한 크롤링이라고 함
- 무한 크롤링은 해당 웹 사이트의 자원을 독점하게 되어 타인의 사용을 막게 되며 웹 사이트에 부하를 주게 됨
- 일부 웹 사이트에서는 동일한 IP로 쉬지 않고 크롤링을 할 경우 접속을 막아 버리는 경우도 있음 (예: 인스타그램)
- 하나의 페이지를 크롤링한 후 1 ~ 2 초 가량 정지하고 다시 다음 페이지를 크롤링하는 것이 좋음
- 신문기사나 책, 사진 등 저작권이 있는 자료를 통해 영업용 이익을 취득하는 행위 저작권법 위반으로 법적 제재를 받을 수 있으므로 유의해야함

로봇 배제 표준

- 로봇 배제 표준이란?
- 웹사이트에 로봇이 접근하는 것을 방지하기 위한 규약. **robots.txt**에 기술하고 있음
- 로봇에 의한 접근이 허용되는 경우라도 웹 서버에 무리가 갈 만큼 반복적으로 웹 페이지를 요청하는 것과 같이 서비스 안정성을 해칠 수 있는 행위를 하지 않아야 함
- 크롤링(또는 스크래핑)으로 취득한 자료를 임의로 배포하거나 변경하는 등의 행위는 저작권을 침해할 수 있으므로 저작권 규정을 준수해야 함

로봇 배제 표준

템플릿 태그	설 명
User-agent: * Disallow: /	모든(*) 로봇에게 루트 디렉터리(/) 이하 모든 문서에 대한 접근을 차단한다.
User-agent: * Allow: /	모든(*) 로봇에게 루트 디렉터리(/) 이하 모든 문서에 대한 접근을 허락한다.
User-agent: * allow: /temp/	모든(*) 로봇에게 특정 디렉터리(/temp/)에 대한 접근을 허락한다.

로봇 배제 표준

```
← → ↻ google.com/robots.txt

User-agent: *
Disallow: /search
Allow: /search/about
Allow: /search/static
Allow: /search/howsearchworks
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*
Allow: /?hl=*&gws_rd=ssl$
Disallow: /?hl=*&*&gws_rd=ssl
```

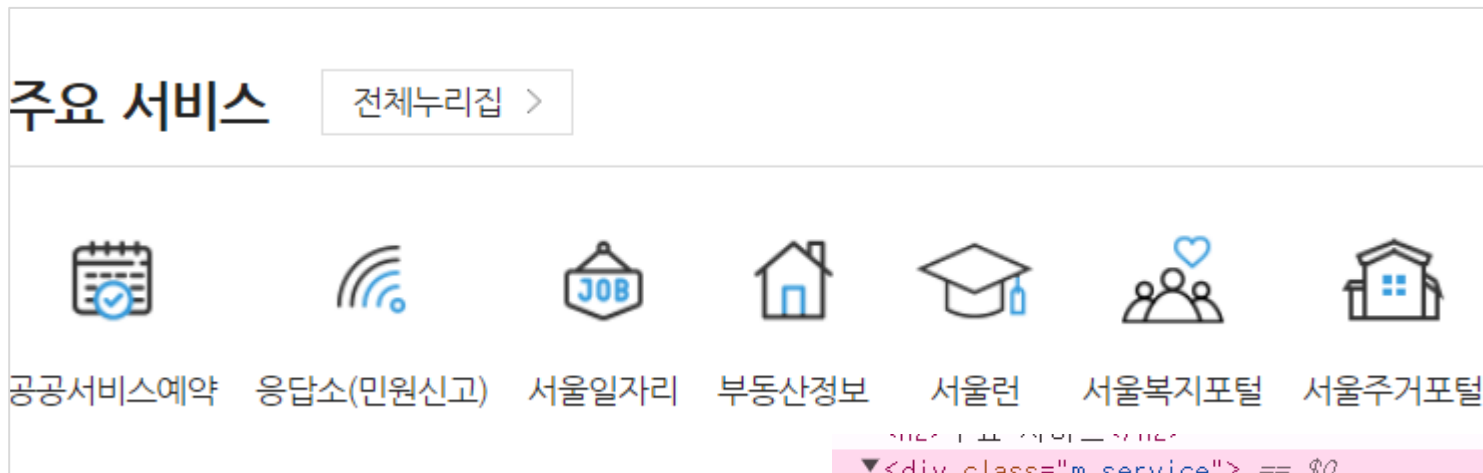
```
← → ↻ news.kbs.co.kr/robots.txt

User-agent: *
Allow: /
Disallow: /resources
Disallow: /api
Disallow: /sokbo
Disallow: /external
Disallow: /preview

Sitemap: https://news.kbs.co.kr/sitemap/recentNewsList.xml
Sitemap: https://news.kbs.co.kr/sitemap/dailyNewsList.xml
Sitemap: https://news.kbs.co.kr/sitemap/danuri.xml
Sitemap: https://news.kbs.co.kr/sitemap/election2024.xml
```

서울시청 웹 크롤링하기

✓ 메뉴 글자 수집하기



```
<div class="m_service"> == $0
  <ul> flex
    <li class="public">
      <a href="//yevak.seoul.go.kr" onclick="action_logging({tr_code: 'serv
        1'})" target="_blank" title="새창">
        <i class="ico_service"></i>
        "공공서비스예약"
      </a>
    </li>
    <li class="answer">
      <a href="//eungdapso.seoul.go.kr" onclick="action_logging({tr_code: '
        1'})" target="_blank" title="새창">
```

서울시청 웹 크롤링하기

✓ 메뉴 글자 수집하기

```
import requests
from bs4 import BeautifulSoup

url = "https://www.seoul.go.kr/main/index.jsp"
response = requests.get(url)
html = BeautifulSoup(response.text, 'html.parser')
# print(html)

# html.select('a')
first_li = html.select_one('li.public')
print(first_li)
```

✓ 0.1s

```
<li class="public">
<a href="//yeyak.seoul.go.kr" onclick="action_logging({tr_code:
</li>
```

서울시청 웹 크롤링하기

✓ 메뉴 글자 수집하기

```
lis = html.select('div.m_service > ul > li')
# print(lis)

for li in lis:
    print(li.text)

# 인덱싱
print(lis[1].text)
print(lis[-1].text)
```

✓ 0.0s

공공서비스예약

응답소(민원신고)

실습1. 국립중앙박물관 관람 정보

- 1) 국립중앙박물관 사이트에 접속한다.
- 2) robots.txt를 확인한다.
- 3) 관람시간과 관람료를 크롤링한다.

🕒 **관람시간** 월/화/목/금/일 10:00 ~ 18:00 수/토 10:00 ~ 21:00 * 입장 마감은 폐관30분 전까지

📄 **관람료** 무료 특별전시는 유료

KBS 뉴스 기사

```
import requests
from bs4 import BeautifulSoup

url = 'https://news.kbs.co.kr/news/pc/view/view.do?ncd=8037507'
response = requests.get(url)
# print(response)
# print(response.text)
html = BeautifulSoup(response.text, 'html.parser')
print(html)
```

✓ 0.3s

KBS 뉴스 기사

```
# title = html.find('h4', class_='headline-title')
title = html.select_one('h4.headline-title')
title
print(title.text)

# detail = html.find('div', class_ = 'detail-body font-size')
detail = html.select_one('div.detail-body')
print(detail.text.strip())
```

✓ 0.0s

전국에 가끔 구름...오늘도 폭염 속 소나기
오늘 전국에 가끔 구름 많겠고 제주도는 대체로 흐리겠습니다. 아침최저기온은 22~27도,

실습2. 전자 신문 메인 기사 크롤링

- 1) 전자 신문 사이트에 접속한다.
- 2) robots.txt를 확인한다.
- 3) 메인 화면 기사를 크롤링한다.
 - (1) 제목 가져오기
 - (2) 발행일 가져오기
 - (3) 본문 내용 가져오기

실습2. 전자 신문 메인 기사 크롤링

삼성 패키징 조직 통합...AVP 개발, TSP 총괄로 이관

발행일 : 2024-08-19 14:21

삼성전자가 반도체 패키징 경쟁력 강화를 위해 흩어진 인력을 통합하는 조직개편을 단행했다. 구현할 첨단 패키징과 기존 패키징 기술을 하나로 묶어 시너지를 극대화하려는 의도로 풀이된다.

19일 업계에 따르면 삼성전자 반도체 사업 부문(DS)은 부문장 직속에 있던 '첨단 패키징(AVP)', '테스트앤시스템패키지(TSP)' 총괄 내로 이관했다. AVP 개발실장이었던 최경세 부사장과 관련 TSP 총괄 부사장 산하로 이동했다.

AVP 개발실은 2.5차원(D)와 3D 패키징 등 첨단 패키징 기술 개발을 전담하는 조직이다. 반도체 경쟁력을 극복하기 위해 첨단 패키징 기술이 급부상하자 경계현 사장이 DS 부문장으로 있던 2021년 만들었다. 경 사장은 여기서 그치지 않고 첨단 패키징을 D램이나 시스템 반도체처럼 사업화를 사업 관련 인력까지 배치해 'AVP사업팀'을 만들었다.

명언 크롤링

- Quotes to Scrape

quotes.toscrape.com

Quotes to Scrape

"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking."
by [Albert Einstein](#) (about)
Tags: [change](#) [deep-thoughts](#) [thinking](#) [world](#)

"It is our choices, Harry, that show what we truly are, far more than our abilities."
by [J.K. Rowling](#) (about)
Tags: [abilities](#) [choices](#)

```

▼<div class="quote" itemscope itemtype="http://schema.org/CreativeWork">
  ▼<span class="text" itemprop="text">
    ▼<font style="vertical-align: inherit;">
      <font style="vertical-align: inherit;">우리가 만든 세상은 우리의 사고
      다. 우리의 사고를 바꾸지 않고는 세상을 바꿀 수 없습니다.</font>
    </font>
  </span>
  ▼<span>
    ▶<font style="vertical-align: inherit;">...</font>
    ▼<small class="author" itemprop="author">
      ▼<font style="vertical-align: inherit;">
        <font style="vertical-align: inherit;">아인슈타인 </font>
      </font>
    </small>
    ▼<a href="/author/Albert-Einstein">
      ▼<font style="vertical-align: inherit;">
        <font style="vertical-align: inherit;">(정보)</font>
      </font>
    </a>
  </span>
  ▶<div class="tags">...</div>

```

명언 크롤링

- Quotes to Scrape

```
import requests as req
from bs4 import BeautifulSoup

url = "https://quotes.toscrape.com/"
res = req.get(url)

html = BeautifulSoup(res.text, 'html.parser')
# print(html)

# 명언(find_all)
quote_div = html.find_all('div', class_='quote')
# print(quote_div)
quote_div_span = html.find_all('span', class_='text')
# print(quote_div_span)
# print(len(quote_div_span))
# 리스트 내포
# [i.find_all('span', class_='text')[0].text for i in quote_div]
```

명언 크롤링

```
# 명언 추출(select)
quote_text = html.select('div.quote > span.text')
# print(quote_text)
# print(len(quote_text))
# print(quote_text[0].text)

# for quote in quote_text:
#     print(quote.text)

# 리스트 내포
[i.text for i in quote_text]

# 명언 말한 사람
quote_author = html.select('div.quote > span > small.author')
# print(quote_author)
[i.text for i in quote_author]

# 말한 사람 정보 링크
quote_link = html.select('div.quote > span > a')
# print(quote_link[0].attrs)
# print(quote_link[0]['href'])
[i['href'] for i in quote_link]
```

```
['/author/Albert-Einstein',
 '/author/J-K-Rowling',
 '/author/Albert-Einstein',
 '/author/Jane-Austen',
 '/author/Marilyn-Monroe',
 '/author/Albert-Einstein',
 '/author/Andre-Gide',
 '/author/Thomas-A-Edison',
 '/author/Eleanor-Roosevelt',
 '/author/Steve-Martin']
```

실습3. 환율정보 크롤링하기

- 네이버 > 증권 > 시장지표 > 환전 고시 환율에 접속하여,
아래 출력결과 이미지대로 크롤링 하세요

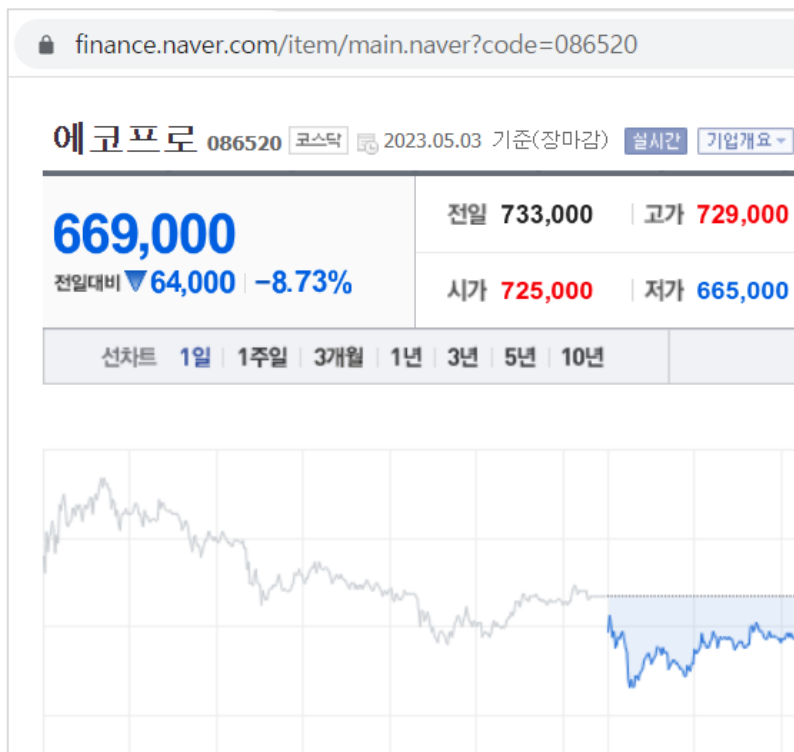


USD 1,335.00
JPY(100엔) 916.11
EUR 1,484.32
CNY 187.25

실습4. 주식정보 크롤링하기

◆ 주식 정보 가져오기

네이버 > 증권(금융 홈) > 주식 종목(우측 하단)



```
<div class="rate_info">
  <div class="today">...</div> == $0
  <table summary="주요 시세(전일종가, 시고저가, 거래량, ...)">
  </table>
  <div class="chart">...</div>
  <a href="javascript:fchartStatus.showChartArea('005930',
lose" onclick="clickcr(this,'sop.toggle','','',event);">
</div>
```

복.습.철.저

수고하셨습니다