

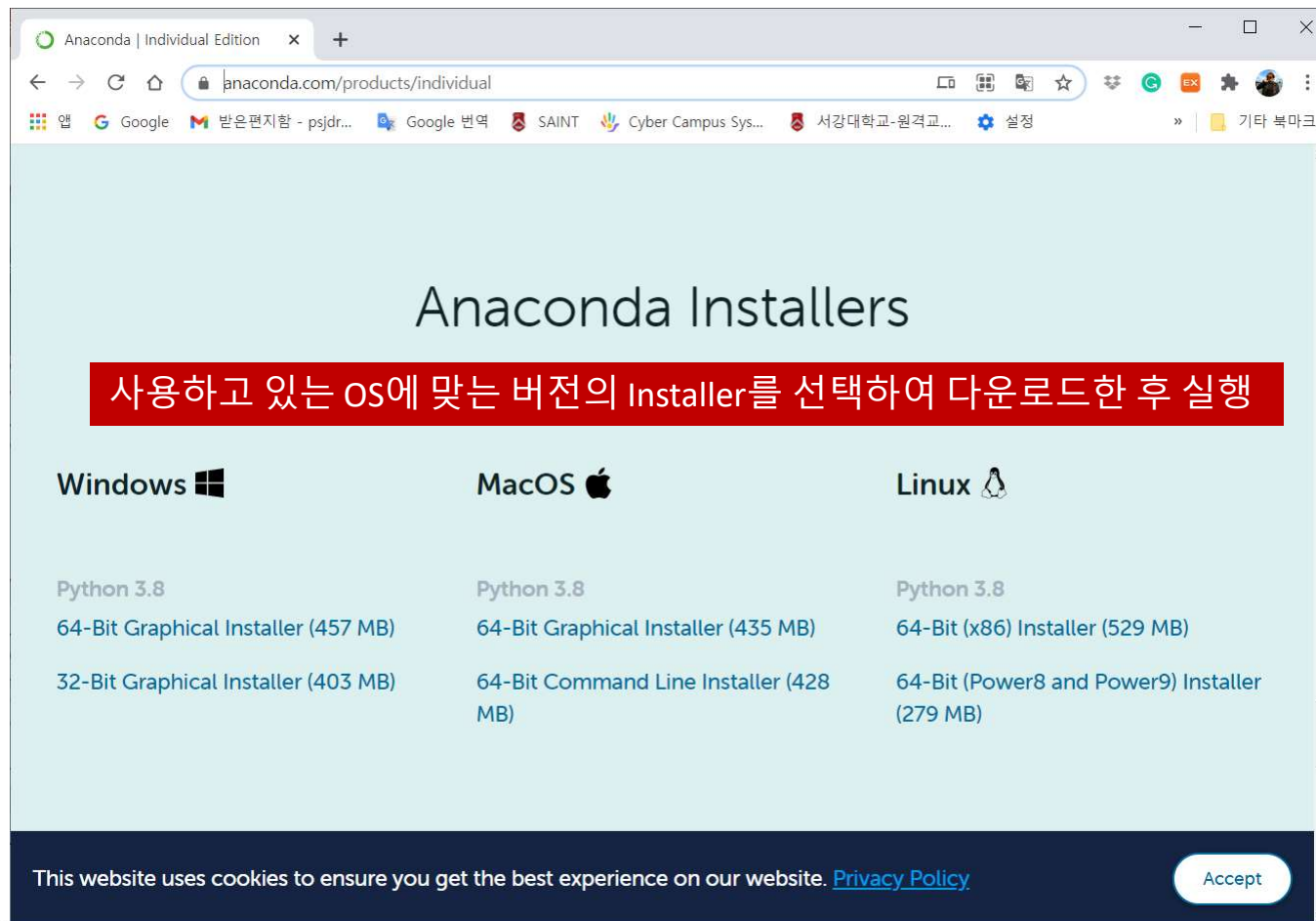
데이터 분석에 필요한 개발환경 구축

Soojin Park

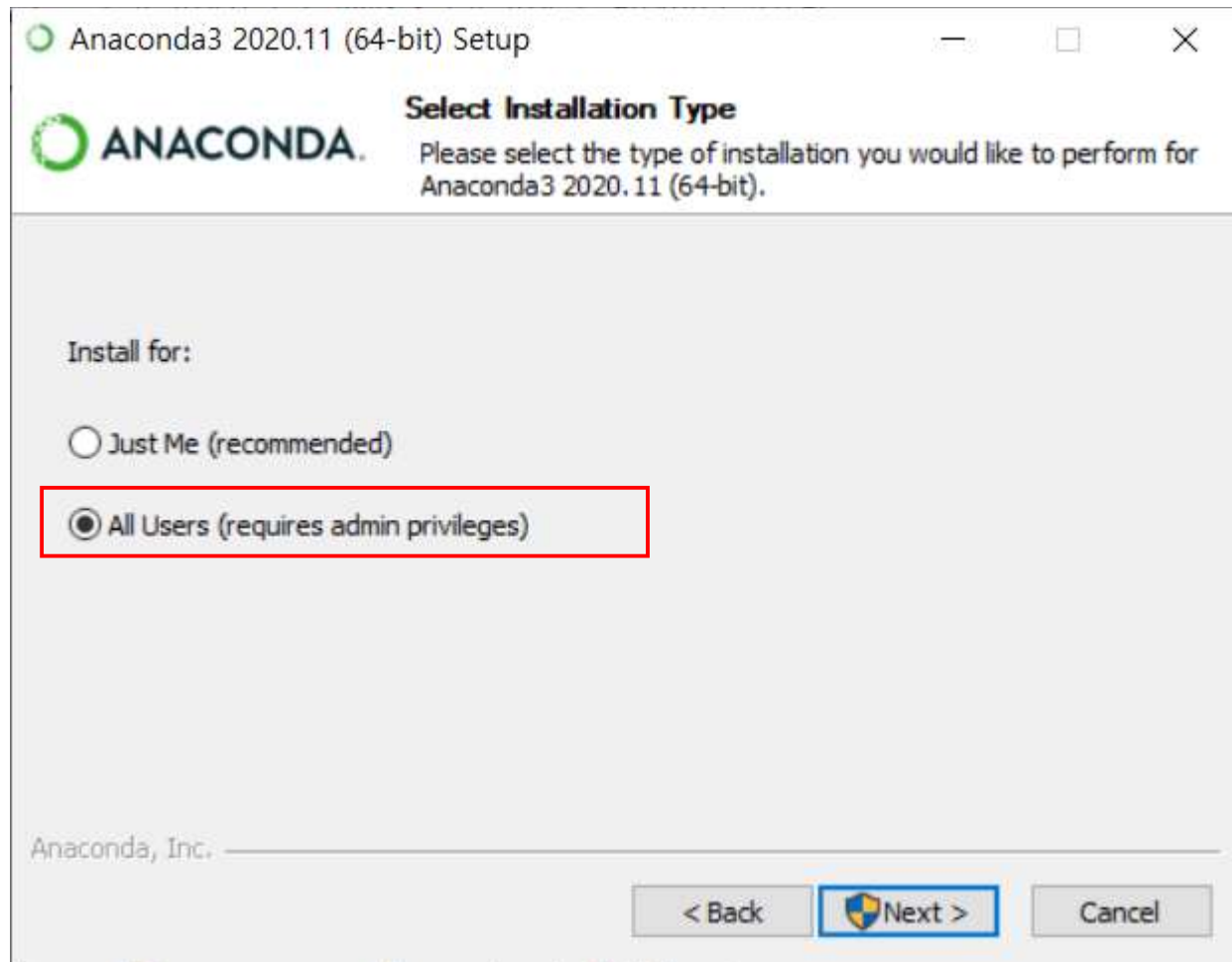
psjdream@sogang.ac.kr

Anaconda 설치

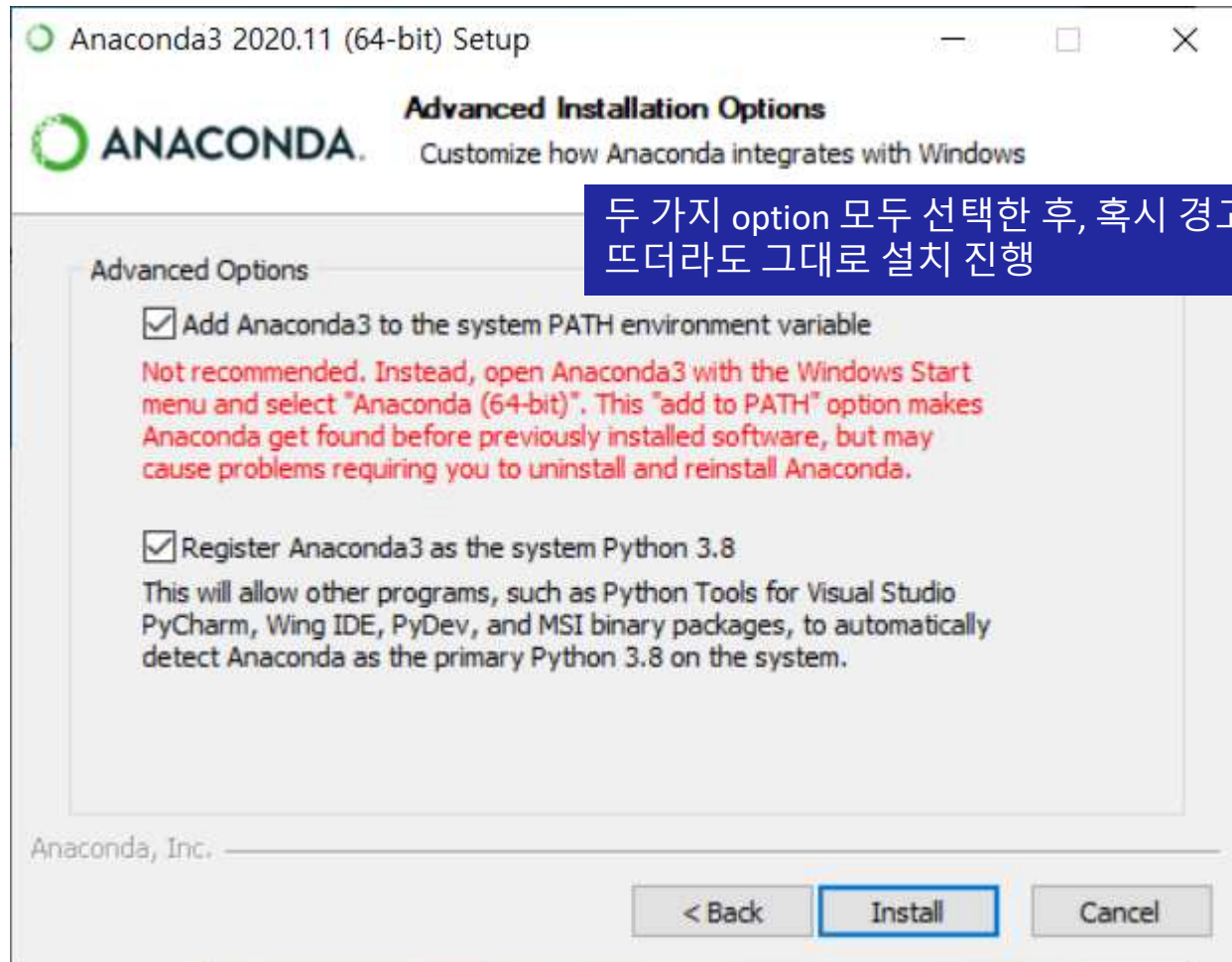
- 다운로드 site: <https://www.anaconda.com/products/individual>
- 설치 및 실행법 : https://www.youtube.com/watch?v=Pm_S2cjdeFI



Anaconda 설치



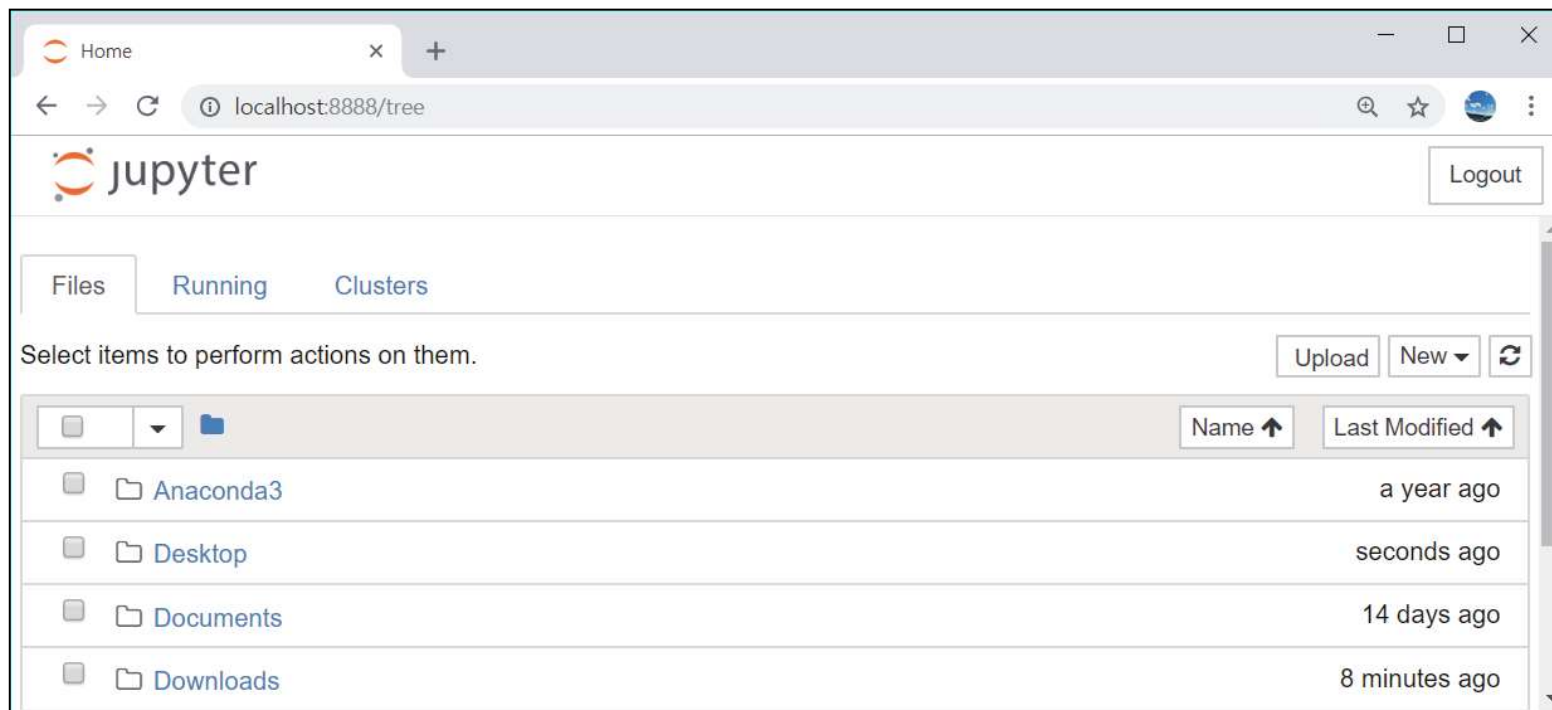
Anaconda 설치



두 가지 option 모두 선택한 후, 혹시 경고 메시지가 뜨더라도 그대로 설치 진행

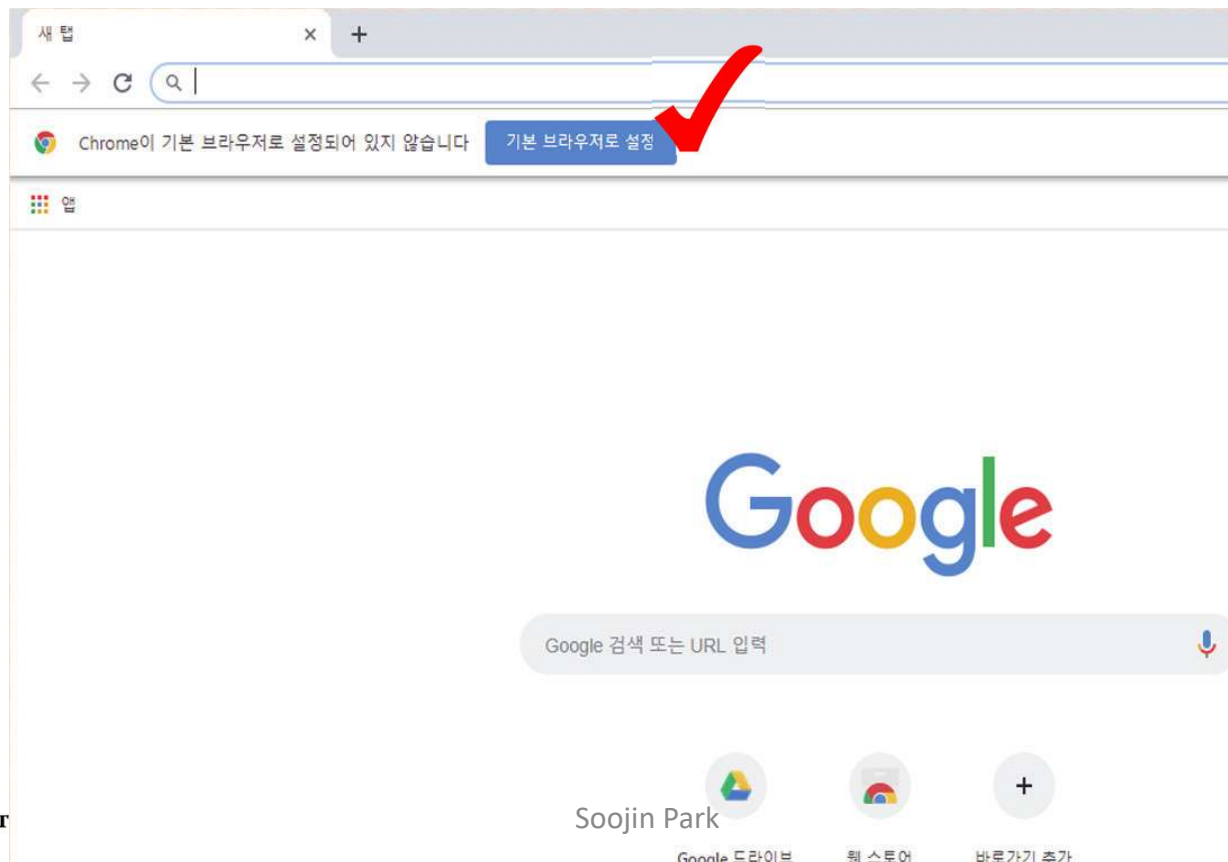
Jupyter Notebook 실행하기

- 주피터 노트북 실행하기
 - Window command 창에 Jupyter Notebook 입력한 후, 검색하여 [열기] 실행
- 주피터 노트북 사용법: <https://www.youtube.com/watch?v=nLDyIDLR1VU>



데이터 분석에 필요한 환경 만들기

- 크롬 브라우저를 기본 브라우저로 설정
 - Jupiter notebook이 크롬 브라우저 환경 최적화 되어 있기 때문



데이터 분석에 필요한 환경 만들기

- 크롬 브라우저를 기본 브라우저로 설정하기 싫다면

Jupyter 실행시킨 후 뜨는 console 창에 있는 웹주소를 복사하여, 크롬 주소창에 입력

```
선택 Jupyter Notebook (Anaconda3) (1)
[I 22:04:16.611 NotebookApp] Writing notebook server cookie secret to C:\Users\ttang\AppData\Roaming\notebook_cookie_secret
[I 22:04:18.280 NotebookApp] JupyterLab extension loaded from C:\Users\ttang\Anaconda3\lib\site-packages
[I 22:04:18.280 NotebookApp] JupyterLab application directory is C:\Users\ttang\Anaconda3\share\jupyterlab
[I 22:04:18.284 NotebookApp] Serving notebooks from local directory: C:\Users\ttang
[I 22:04:18.284 NotebookApp] The Jupyter Notebook is running at:
[I 22:04:18.284 NotebookApp] http://localhost:8888/?token=b0aa79a3ac5172e8e990f097b6cb62af65a44a23ad68858f
[I 22:04:18.285 NotebookApp] or http://127.0.0.1:8888/?token=b0aa79a3ac5172e8e990f097b6cb62af65a44a23ad68858f
[I 22:04:18.285 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to
[C 22:04:18.385 NotebookApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/ttang/AppData/Roaming/jupyter/runtime/nbserver-9396-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=b0aa79a3ac5172e8e990f097b6cb62af65a44a23ad68858f
    or http://127.0.0.1:8888/?token=b0aa79a3ac5172e8e990f097b6cb62af65a44a23ad68858f
[I 22:04:28.287 NotebookApp] Creating new notebook in
[I 22:04:28.308 NotebookApp] Writing notebook-signing key to C:\Users\ttang\AppData\Roaming\jupyter\notebook-signing-key
[I 22:04:30.041 NotebookApp] Kernel started: 390ccefd-cb4d-4ca2-8eba-c5bc7ac75d29
```

기온 데이터 분석 및 가시화

기온 공공데이터 살펴보기

- 기상자료개방포털 홈페이지(<https://data.kma.go.kr>)를 통해 기상 관련 데이터 수집

기상자료개방포털

'관측'을 검색하세요



인기검색어



기상자료개방포털이란?

데이터

기후통계분석

간행물

소통과 참여



데이터
전체보기

지도로 찾기

관측

예·특보

대용량

기상위성

수치모델

기상레이더

날씨!
데이터가 되다

OPEN
API

지진화산

기후통계

기온 공공데이터 살펴보기

- 기상자료개방포털 홈페이지 - [기후통계분석] - [통계분석] - [기온분석]

기상자료개방포털

'관측'을 검색하세요



인기검색어



기상자료개방포털이란?

데이터

기후통계분석

간행물

소통과 참여



> 평년값

- 우리나라 기후평년값
- 세계기후평년값

> 통계분석

- 조건별통계
- 기온분석
- 강수량분석
- 다중지점통계
- 24절기
- 순위값
- 장마

> 기상현상일수

- 강수일수
- 눈일수
- 황사일수
- 폭염일수
- 열대야일수
- 안개일수
- 서리일수
- 결빙일수
- 우박일수
- 폭풍일수

> 계급별일수

- 전운량 계급별일수
- 강수 계급별일수
- 바람 계급별일수(바람장미)

기온 공공데이터 살펴보기

- 검색조건 설정

tip) 기간은 1904년부터 검색이 가능하지만, 실제 데이터는 1907년 10월 1일부터 제공(서울 기준)



기후통계분석

- 평년값
- 통계분석**
- 조건별통계
- 기온분석**
- 강수량분석
- 다중지점통계
- 24절기
- 순위값
- 장마
- 기상현상일수

Home > 기후통계분석 > 통계분석 > 기온분석

기온분석 - 그래프

그래프분포도

자료설명

지점별로 기온의 시계열 분석을 확인합니다.
일, 월, 연의 평균기온, 최저기온, 최고기온을 각각 조회할 수 있습니다.

* (그래프) 평균최고(최저)기온: 일최고(최저)기온의 월평균

* '지역/지점'의 '지역'은 전국 및 광역 단위의 평균 제공(1973년~) (전국 및 광역별 평균에 사용된 지점은 전국 평균산출에 사용되는 45개 지점이며, 제주도는 제주시와 서귀포시 자료임)

검색조건

자료구분일

자료형태기본

기간19040101 ~ 20190117

지역/지점서울

선택

> 검색



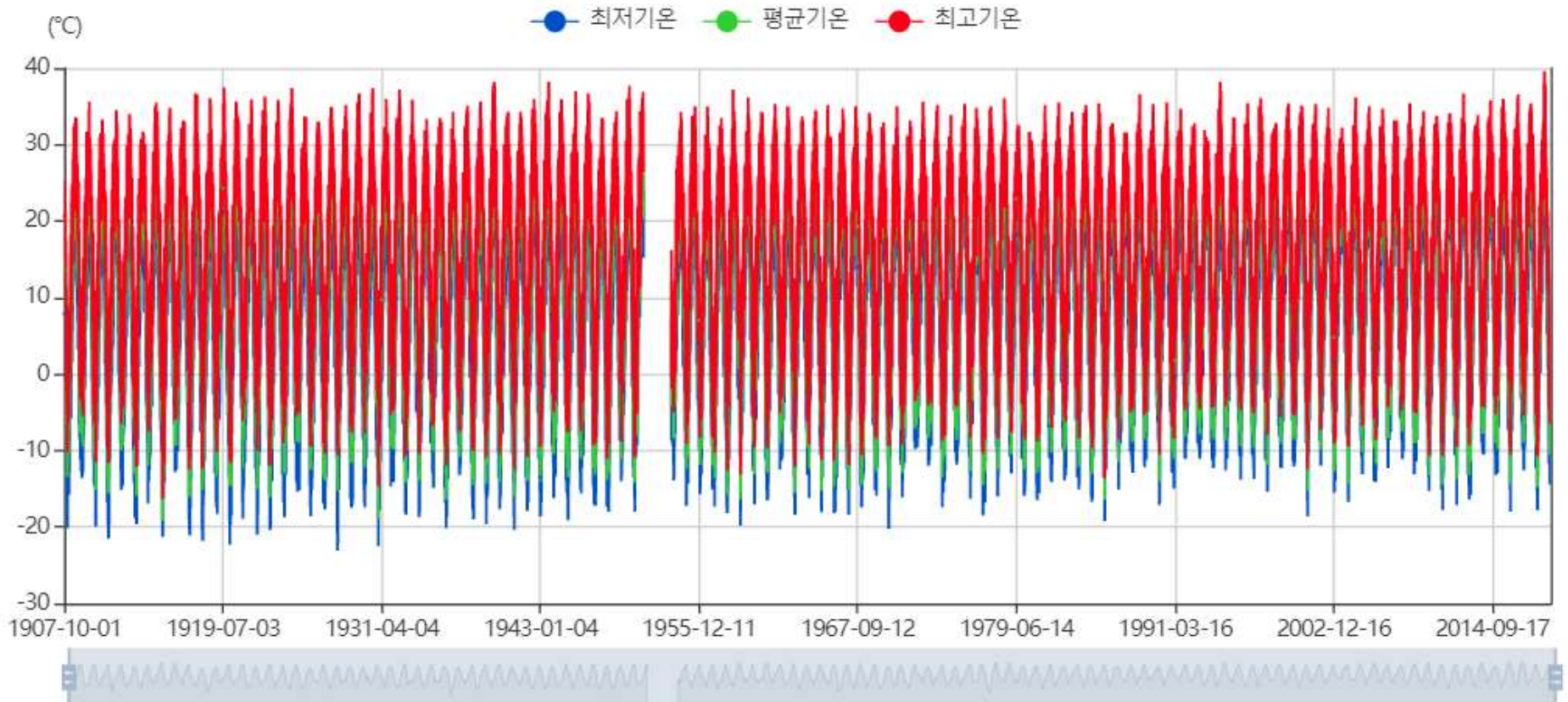
기온 공공데이터 살펴보기

- [CSV다운로드] 버튼 클릭

CSV

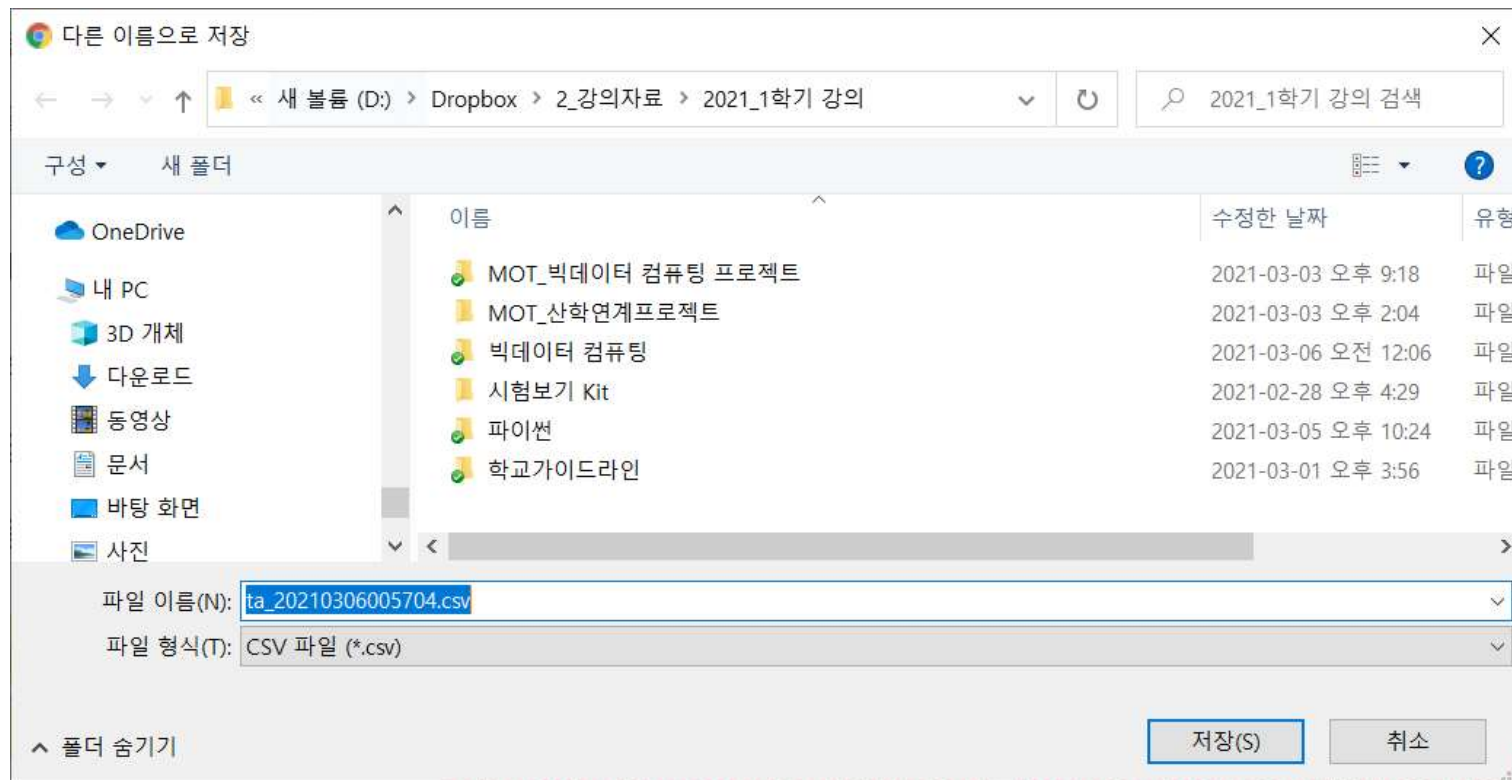
Excel

기온분석 기본 서울(108) 일자료 기간: 19040101 ~ 20190117



기온 공공데이터 살펴보기

- csv 파일을 원하는 폴더에 저장



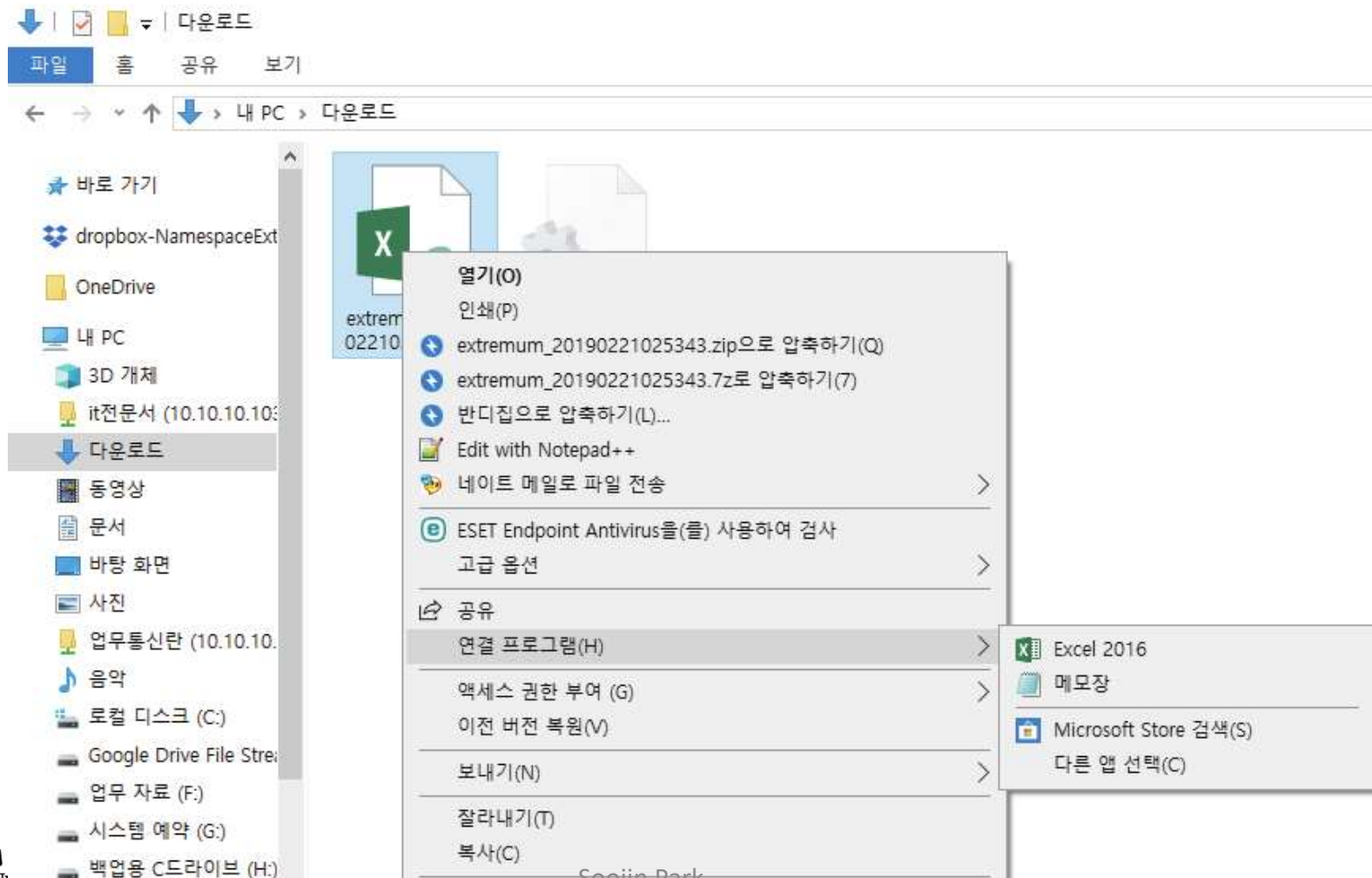
CSV 파일이란

- ' Comma - Separated Values '의 약자
- 각각의 데이터 값을 콤마(,)로 구분하는 파일 형식

1	기온분석					
2	[검색조건]					
3	자료구분 : 일					
4	자료형태 : 기본					
5	지역/지점 : 서울					
6	기간 : 19040101~20190117					
7						
8	날짜	지점	평균기온(°C)	최저기온(°C)	최고기온(°C)	
9	1907-10-01	108	13.5	7.9	20.7	
10	1907-10-02	108	16.2	7.9	22	
11	1907-10-03	108	16.2	13.1	21.3	
12	1907-10-04	108	16.5	11.2	22	
13	1907-10-05	108	17.6	10.9	25.4	
14	1907-10-06	108	13	11.2	21.3	

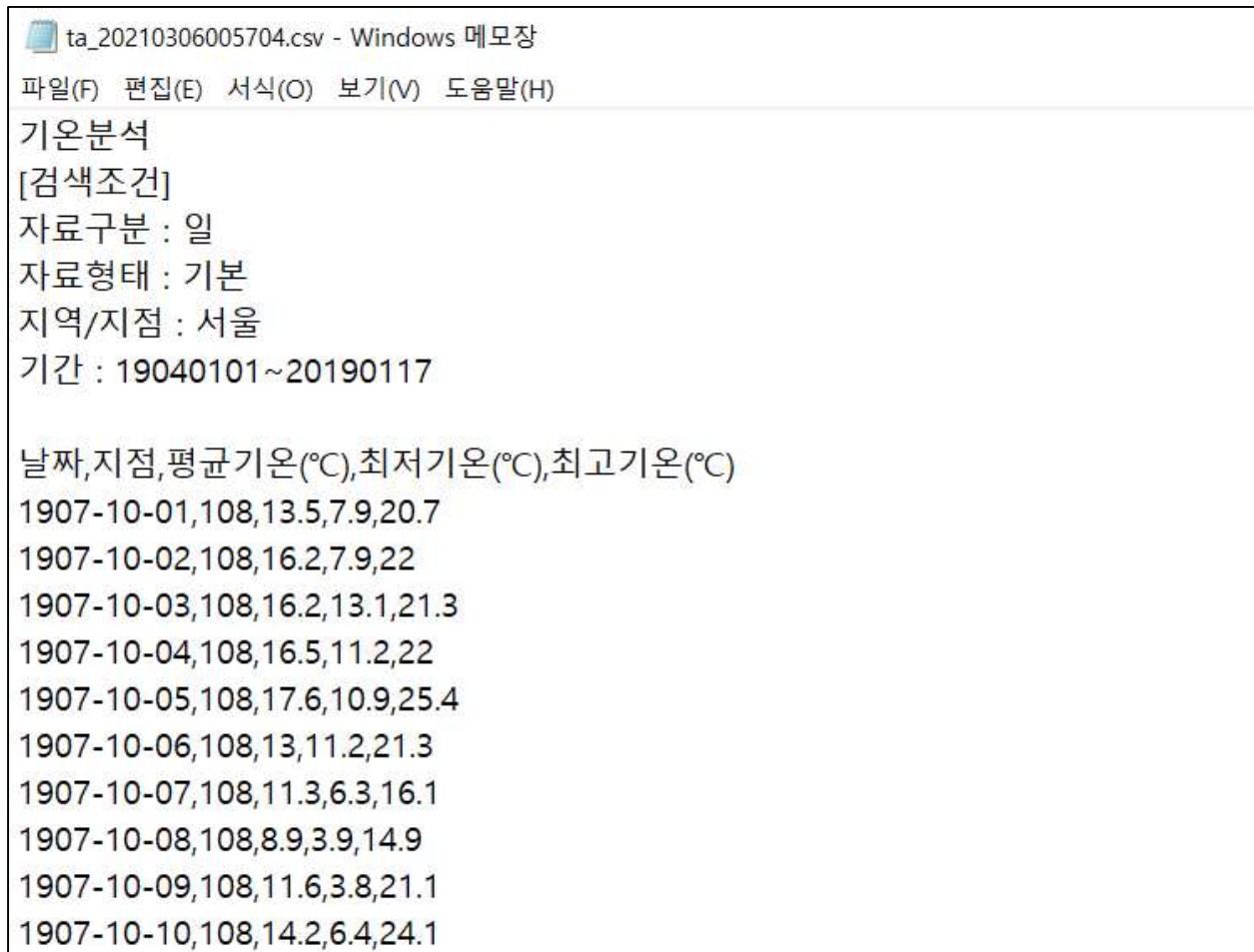
CSV 파일이란

- csv 파일 메모장으로 열어보기



CSV 파일이란

- csv 파일 메모장으로 열어보기



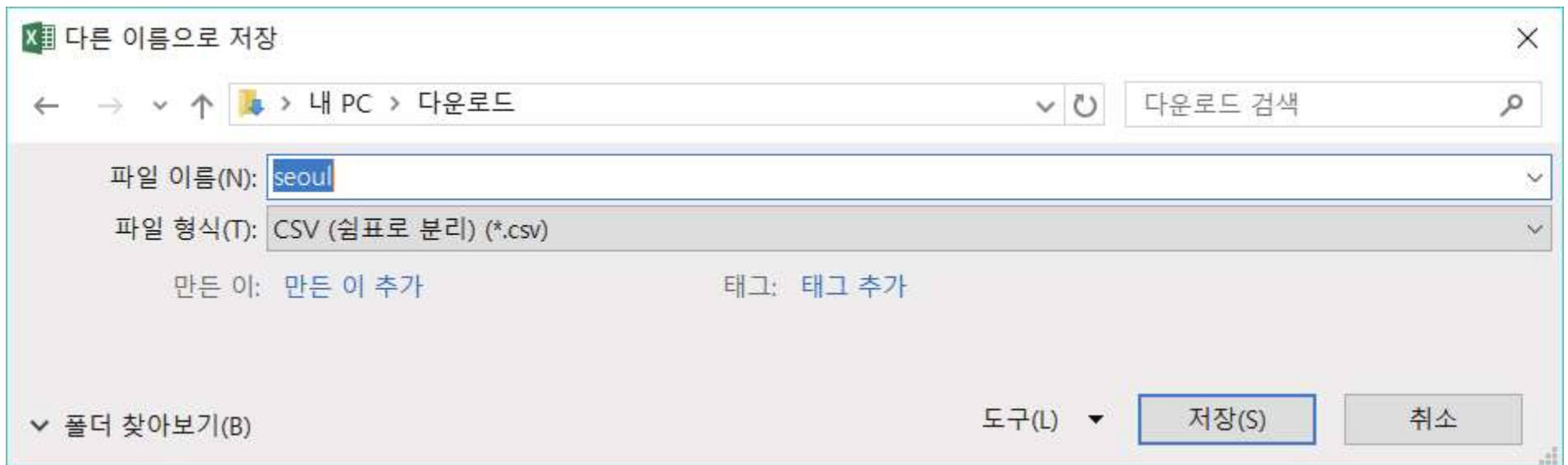
CSV 파일에서 데이터 읽어오기

- 불필요한 데이터(1~7행) 삭제: Excel로 열어서 해당 행 삭제

	A	B	C	D	E
1	기온분석				
2	[검색조건]				
3	자료구분 : 일				
4	자료형태 : 기본				
5	지역/지점 : 서울				
6	기간 : 19040101~20190117				
7					
8	날짜	지점	평균기온(°C)	최저기온(°C)	최고기온(°C)
9	1907-10-01	108	13.5	7.9	20.7
10	1907-10-02	108	16.2	7.9	22
11	1907-10-03	108	16.2	13.1	21.3
12	1907-10-04	108	16.5	11.2	22
13	1907-10-05	108	17.6	10.9	25.4
14	1907-10-06	108	13	11.2	21.3
15	1907-10-07	108	11.3	6.3	16.1

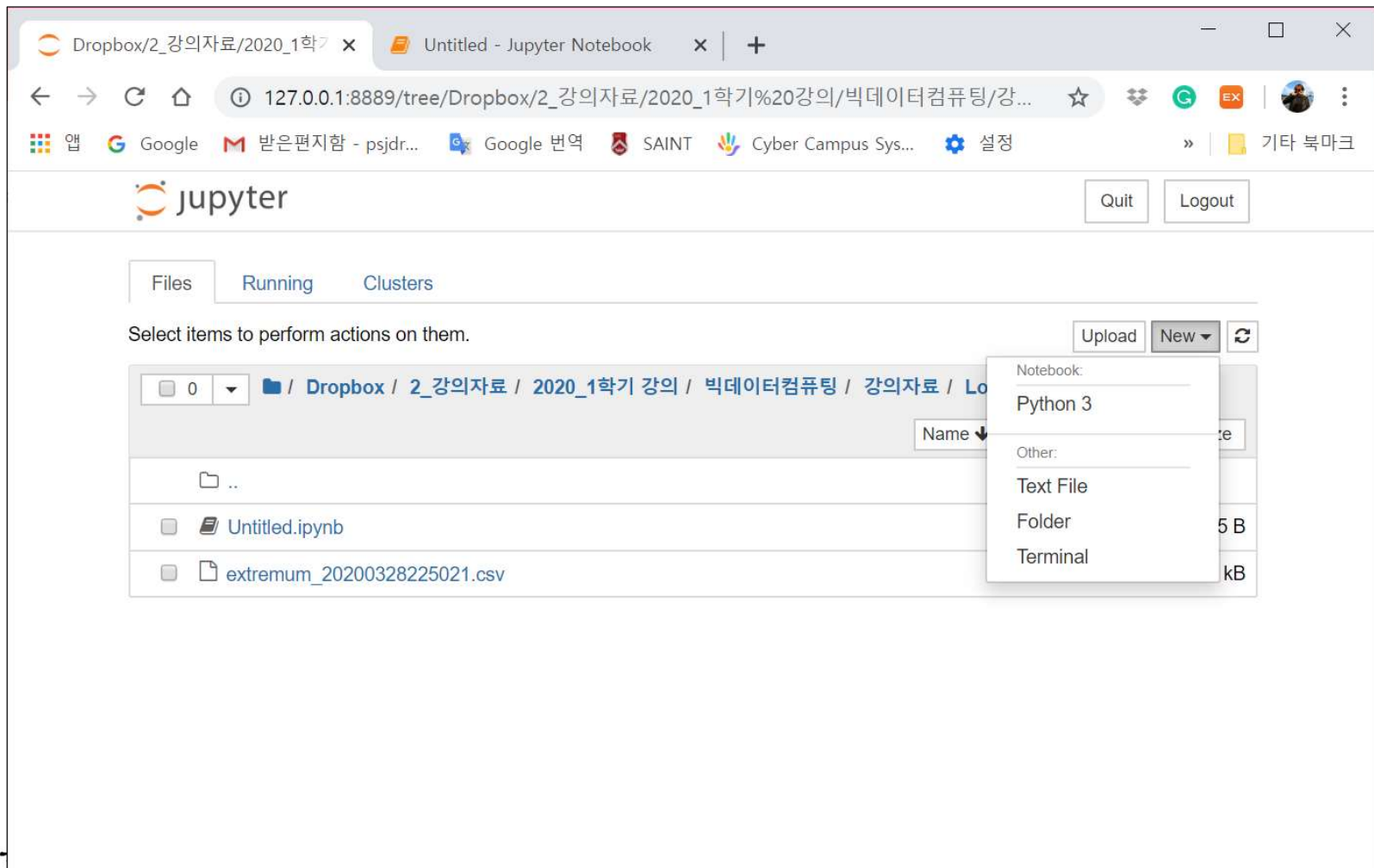
CSV 파일에서 데이터 읽어오기

- [다른 이름으로 저장] – [파일 형식] – [CSV] – [저장]



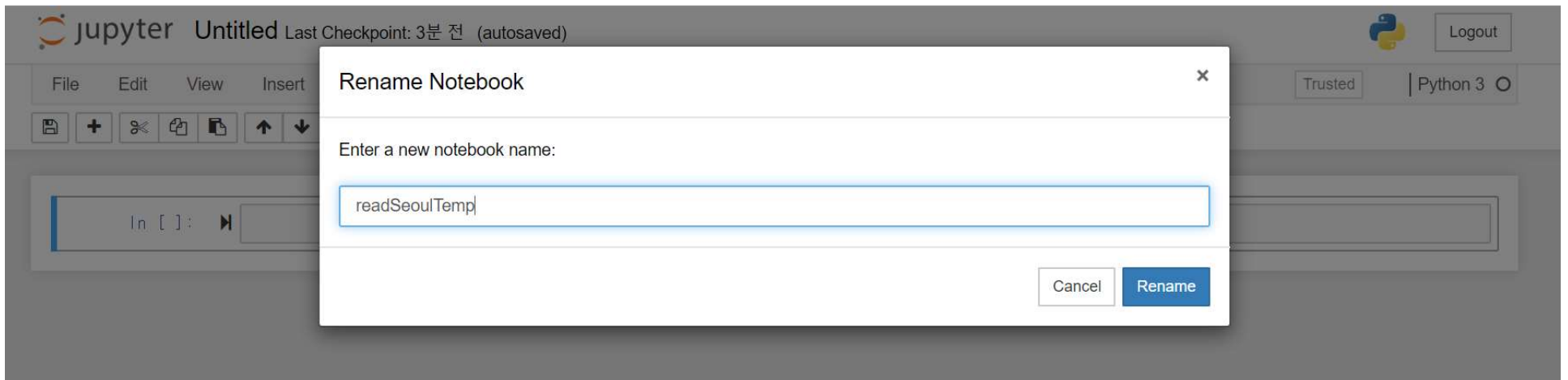
CSV 파일에서 데이터 읽어오기

- seoul.csv 파일 읽어오는 Python code 작성을 위해 새로운 Python 파일 생성



CSV 파일에서 데이터 읽어오기

- 작성한 Python Jupyter Notebook file(.ipynb)파일 이름 변경



CSV 파일에서 데이터 읽어오기

- seoul.csv 파일 읽어와서 print하는 코드 작성

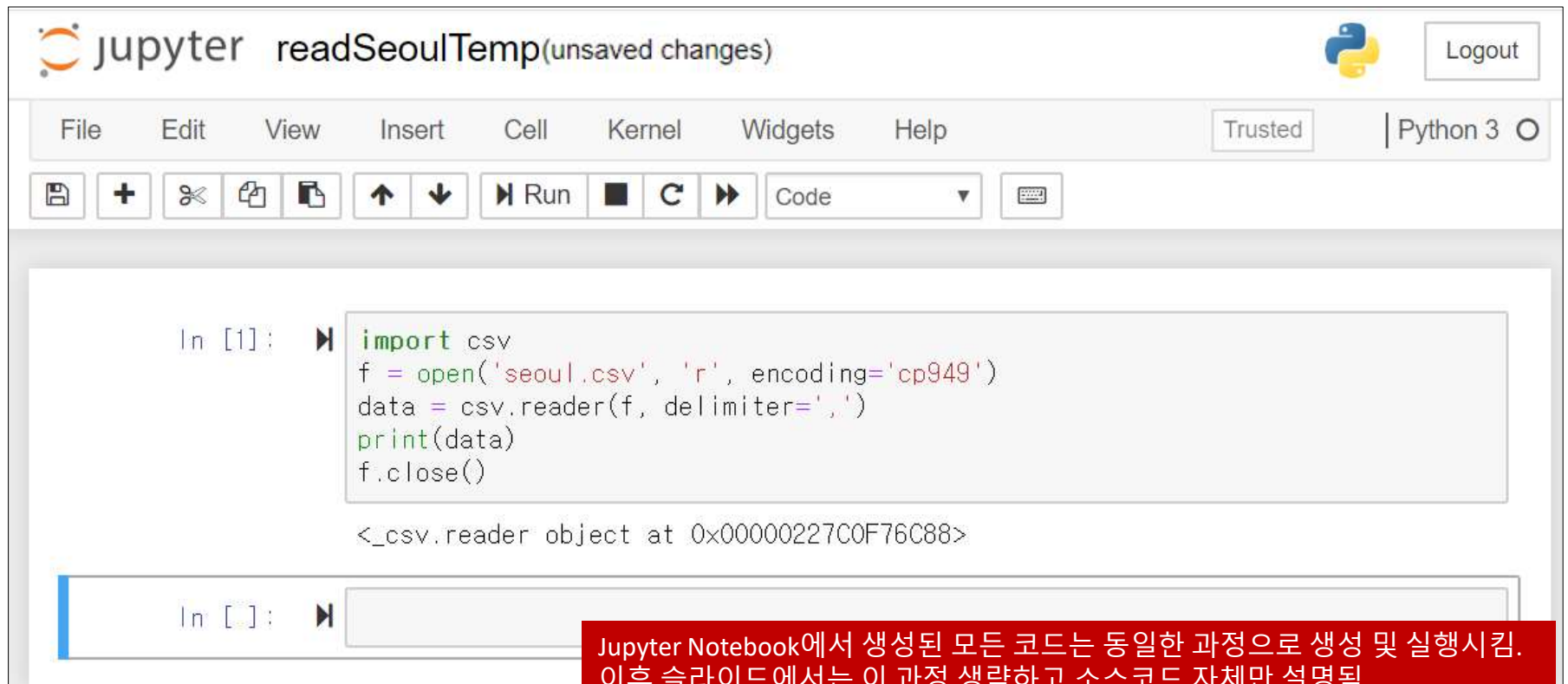
이때, seoul.csv 파일은 이 코드 파일과 동일한 폴더에 위치하여야 함
그렇지 않은 경우, FileNotFoundError 발생

The screenshot shows a Jupyter Notebook interface in a web browser. The browser's address bar displays the URL `127.0.0.1:8889/notebooks/Dropbox/2_강의자료/2020_1학기%...`. The Jupyter interface includes a top bar with the Jupyter logo, the notebook name `readSeoulTemp(unsaved changes)`, and a `Logout` button. Below this is a menu bar with `File`, `Edit`, `View`, `Insert`, `Cell`, `Kernel`, `Widgets`, and `Help`. A toolbar contains icons for file operations and execution. The main area features a code cell with the following Python code:

```
In [ ]: import csv
        f= open('seoul.csv', 'r', encoding = 'cp949')
        data = csv.reader(f, delimiter = ',')
        print(data)
        f.close()
```

CSV 파일에서 데이터 읽어오기

- 파이썬 코드 실행: Run 버튼 클릭 or Ctrl+Shift



The screenshot shows a Jupyter Notebook window titled "readSeoulTemp(unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a code editor. The code in the first cell is as follows:

```
In [1]: ▶ import csv
f = open('seoul.csv', 'r', encoding='cp949')
data = csv.reader(f, delimiter=',')
print(data)
f.close()
```

Below the code, the output is displayed: `<_csv.reader object at 0x00000227C0F76C88>`. The second code cell is currently empty.

Jupyter Notebook에서 생성된 모든 코드는 동일한 과정으로 생성 및 실행시킴.
이후 슬라이드에서는 이 과정 생략하고 소스코드 자체만 설명됨

CSV 파일에서 데이터 읽어오기

```
import csv
f = open('seoul.csv', 'r', encoding='cp949')
data = csv.reader(f, delimiter=',')
print(data)
f.close()
```

①

②

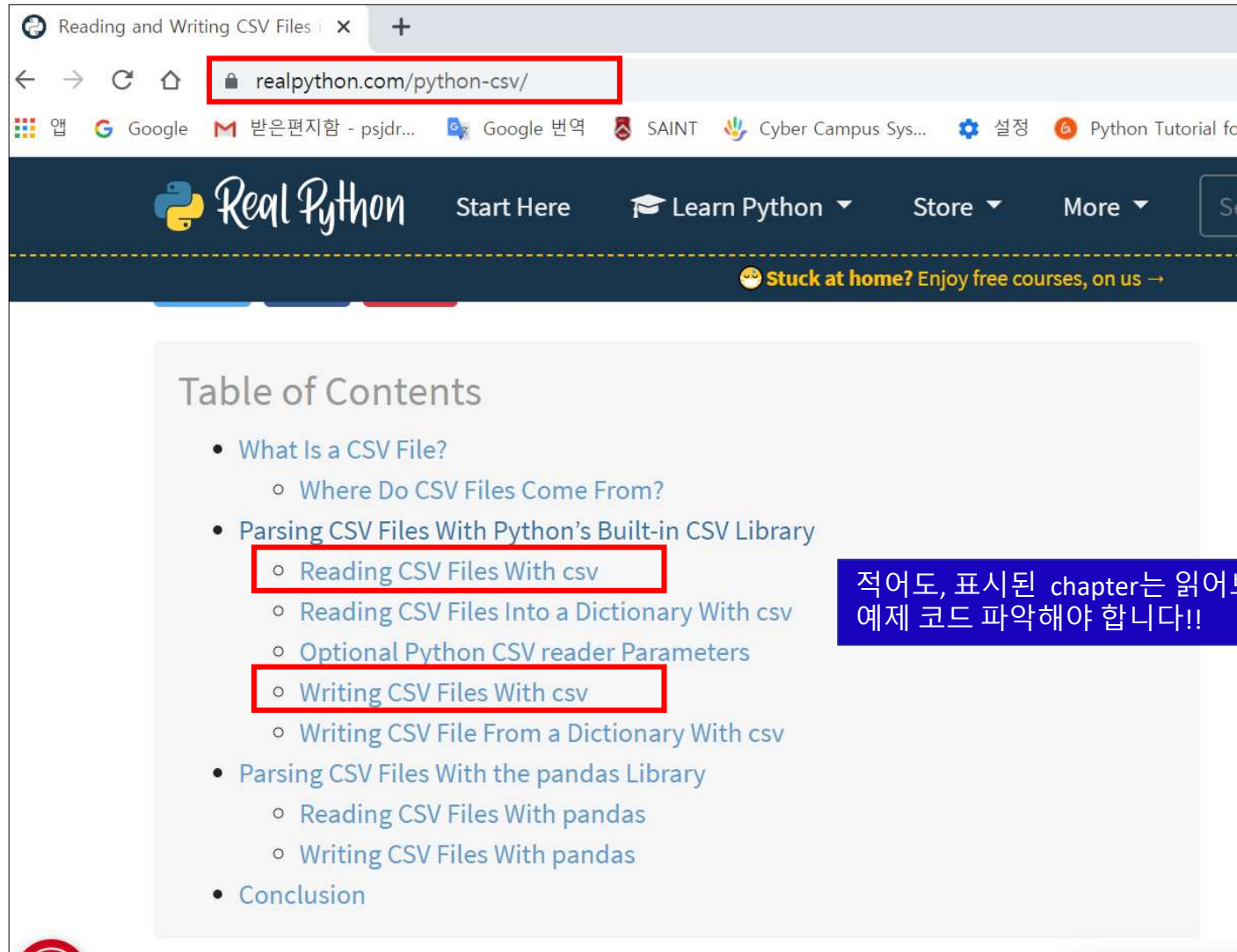
③

④

⑤

- ① csv 모듈을 불러옴
- ② seoul.csv 파일을 read 모드로 읽어옴. 이때 encoding 방식은 cp949(Windows 한글 encoding). 읽어온 파일을 파일 핸들러 f에 할당
- ③ f를 csv 모듈에서 정의하는 reader 함수에 전달하여, data라는 csv reader 객체를 생성. 이때 구분자는 (,)
- ④ Data를 print → data는 객체이기 때문에, 앞 슬라이드의 실행결과와 같이 객체 주소가 출력됨
<_csv.reader object at 0x00000227C0F76C88>
- ⑤ 파일핸들러 f가 가리키는 파일을 닫음

보충공부: Reading/Writing CSV Files with csv



The screenshot shows a web browser window with the URL `realpython.com/python-csv/` highlighted in the address bar. The page title is "Reading and Writing CSV Files". The Real Python logo and navigation links are visible. The main content area displays a "Table of Contents" with the following items:

- What Is a CSV File?
 - Where Do CSV Files Come From?
- Parsing CSV Files With Python's Built-in CSV Library
 - Reading CSV Files With csv
 - Reading CSV Files Into a Dictionary With csv
 - Optional Python CSV reader Parameters
 - Writing CSV Files With csv
 - Writing CSV File From a Dictionary With csv
- Parsing CSV Files With the pandas Library
 - Reading CSV Files With pandas
 - Writing CSV Files With pandas
- Conclusion

적어도, 표시된 chapter는 읽어보고,
예제 코드 파악해야 합니다!!

데이터 출력하기

- 읽어온 seoul.csv 파일의 내용을 프린트하도록 코드 수정

```
import csv
f = open('seoul.csv', encoding='cp949')
data = csv.reader(f)
for row in data :
    print(row)
f.close()
```

→ 4칸 들여쓰기에 주의하세요!

실행
결과

['날짜', '지점', '평균기온(°C)', '최저기온(°C)', '최고기온(°C)']

['1907-10-01', '108', '13.5', '7.9', '20.7']

['1907-10-02', '108', '16.2', '7.9', '22']

['1907-10-03', '108', '16.2', '13.1', '21.3']

['1907-10-04', '108', '16.5', '11.2', '22']

(생략)

→ [] 로 묶여있음

→ List type

→ indexing과 slicing 활용한 데이터 조작이 가능

데이터 출력하기

- 누락된 데이터 확인
 - 1950년 9월 1일: 기온자료 누락 → 6.25 전쟁 당시 데이터 미수집으로 추측
 - 2017년 10월 12일: 최고기온 누락 → 입력과정 오류로 추측
- Big Data Veracity 속성(data는 완전 무결하지 않음을 인정하고 application 구현할 필요가 있음)

실행
결과

```
.....  
[1950-08-29', '108', '23.1', '16.8', '30.4']  
[1950-08-30', '108', '24.6', '18', '32.6']  
[1950-08-31', '108', '25.4', '20.1', '32.5']  
[1950-09-01', '108', ' ', ' ', ' '] → 기온 데이터가 누락됨  
[1950-09-02', '108', ' ', ' ', ' ']  
(생략)  
[2017-10-12', '108', '11.4', '8.8', '']  
....
```

헤더 저장하기

- next() 함수를 활용해 헤더 저장하기

```
import csv
f = open('seoul.csv')
data = csv.reader(f)
header = next(data)
print(header)
f.close()
```

next() : iterator가 가리키는 item을 return하고, 다음 item으로 탐색위치를 이동시킴
→ 파일 f를 읽자마자 next() 함수 호출
→ 파일의 첫번째 행을 return

['날짜', '지점', '평균기온(°C)', '최저기온(°C)', '최고기온(°C)']

https://www.w3schools.com/python/ref_func_next.asp

헤더 저장하기

- 헤더를 제외한 데이터 한 행씩 출력하기

```
import csv
f = open('seoul.csv')
data = csv.reader(f)
header = next(data)
for row in data :
    print(row)
f.close()
```

```
['1907-10-01', '108', '13.5', '7.9', '20.7']
['1907-10-02', '108', '16.2', '7.9', '22']
(생략)
```

기온 공공데이터에 대한 질문 작성

- 데이터에 질문하기
 - 데이터 분석은 내가 관심 있는 데이터에 대한 호기심에서 출발

	A	B	C	D	E
1	날짜	지점	평균기온(°C)	최저기온(°C)	최고기온(°C)
2	1907-10-01	108	13.5	7.9	20.7
3	1907-10-02	108	16.2	7.9	22
4	1907-10-03	108	16.2	13.1	21.3
5	1907-10-04	108	16.5	11.2	22
6	1907-10-05	108	17.6	10.9	25.4
7	1907-10-06	108	13	11.2	21.3
8	1907-10-07	108	11.3	6.3	16.1
9	1907-10-08	108	8.9	3.9	14.9
10	1907-10-09	108	11.6	3.8	21.1
11	1907-10-10	108	14.2	6.4	24.1
12	1907-10-11	108	15.4	10.1	20.4
13	1907-10-12	108	13.9	11.1	17.4

기온 공공데이터에 대한 질문 작성

- **서울이 가장 더웠던 날은 언제였을까?** 얼마나 더웠을까?
- 일교차가 가장 큰 시기는 1년 중 언제쯤일까?
- 겨울에는 언제 가장 추울까? 12월? 1월? 2월?
- 가장 덥다고 알려진 대구보다 서울이 더 더운 날이 1년 중 얼마나 있을까?

질문 다듬기

- 서울이 가장 더웠던 날은 언제였을까? 얼마나 더웠을까?
 - 가장 더웠던 날의 기준은?
 - 우리가 갖고 있는 데이터는?
- > 기상 관측 이래, 서울의 최고 기온이 가장 높았던 날은 언제였고, 몇 도였을까?

문제 해결 방법 구상하기

- 질문 : 기상 관측 이래, 서울의 최고 기온이 가장 높았던 날은 언제였고, 몇 도였을까?
- 질문을 해결하는데 필요한 데이터는?
 - 날짜, 최고기온 데이터

	A	B	C	D	E
1	날짜	지점	평균기온(°C)	최저기온(°C)	최고기온(°C)
2	1907-10-01	108	13.5	7.9	20.7
3	1907-10-02	108	16.2	7.9	22
4	1907-10-03	108	16.2	13.1	21.3
5	1907-10-04	108	16.5	11.2	22
6	1907-10-05	108	17.6	10.9	25.4
7	1907-10-06	108	13	11.2	21.3
8	1907-10-07	108	11.3	6.3	16.1
9	1907-10-08	108	8.9	3.9	14.9
10	1907-10-09	108	11.6	3.8	21.1
11	1907-10-10	108	14.2	6.4	24.1
12	1907-10-11	108	15.4	10.1	20.4
13	1907-10-12	108	13.9	11.1	17.4

문제 해결 방법 구상하기

- 질문 : 기상 관측 이래, 서울의 최고 기온이 가장 높았던 날은 언제였고, 몇 도였을까?
- 질문을 해결하는데 필요한 절차(알고리즘)는?

- 1 | 데이터를 읽어온다.
- 2 | 순차적으로 최고 기온을 확인한다.
- 3 | 최고 기온이 가장 높았던 날짜의 데이터를 저장한다.
- 4 | 최종 저장된 데이터를 출력한다.

파이썬 코드로 구현하기

- **step 1)** 데이터 불러서 한 행씩 출력하기

```
import csv
f = open('seoul.csv')
data = csv.reader(f)
header = next(data)
for row in data :
    print(row)
f.close()
```

```
['1907-10-01', '108', '13.5', '7.9', '20.7']
['1907-10-02', '108', '16.2', '7.9', '22']
(생략)
```

파이썬 코드로 구현하기

- **step 2)** 데이터 중 최고 기온을 실수로 변환하여 한 행씩 출력하기

```
import csv
f = open('seoul.csv')
data = csv.reader(f)
header = next(data)
for row in data :
    row[-1] = float(row[-1])    # 최고 기온을 실수로 변환
    print(row)
f.close()
```

Q1. 왜 실수로 변환해야 할까요??

Basic Python: Unique Indexing in Python

If it traverses backwardly: decrease index by 1



Negative Index [-8] [-7] [-6] [-5] [-4] [-3] [-2] [-1]

Positive Index [0] [1] [2] [3] [4] [5] [6] [7]

primes	2	3	5	7	11	13	17	26
---------------	---	---	---	---	----	----	----	----



If it traverses forwardly: increase index by 1

negative index = positive index – len(list)

`primes[0] == 2 == primes[-8]`

`primes[2] == 5 == primes[-6]`

`primes[-2] == 17 == primes[6]`

Basic Python: Unique Indexing in Python

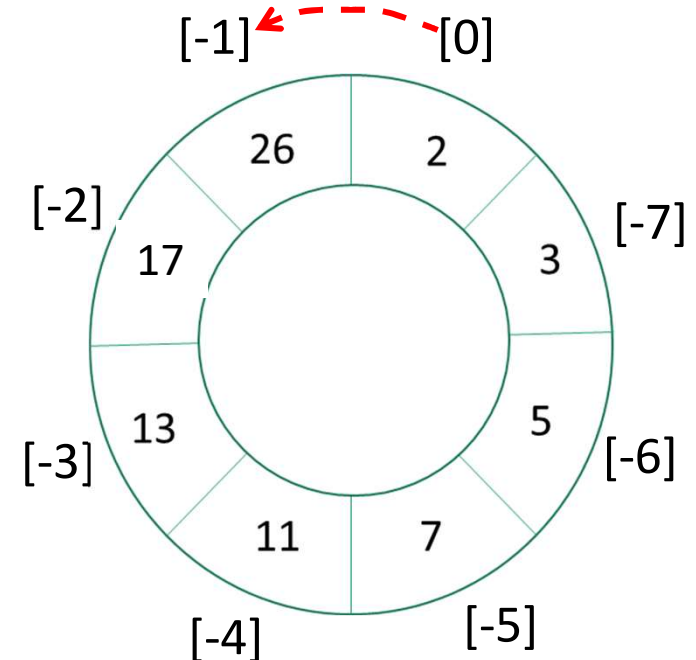
If it traverses backwardly: decrease index by 1

Negative Index	[-8]	[-7]	[-6]	[-5]	[-4]	[-3]	[-2]	[-1]
Positive Index	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]

primes	2	3	5	7	11	13	17	26
---------------	---	---	---	---	----	----	----	----

← traverse backward

The negative index means we start at the end of the list and go left when reading a list.



Basic Python: Unique Indexing in Python

If it traverses backwardly: decrease index by 1

Negative Index	[-8]	[-7]	[-6]	[-5]	[-4]	[-3]	[-2]	[-1]
Positive Index	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]

primes ← 2 3 5 7 11 13 8 26

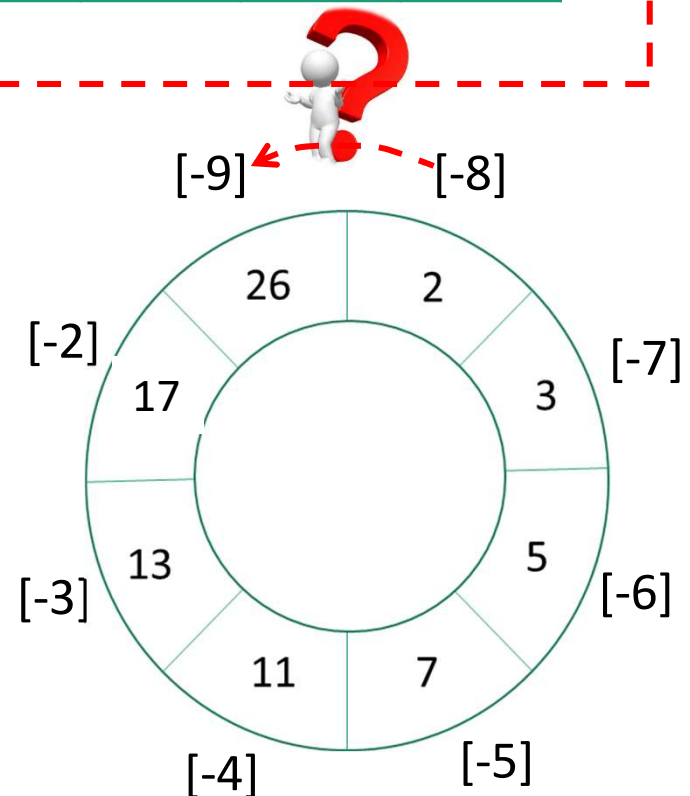
traverse backward

$-len(list) \leq \text{index} < len(list)$

$-len(primes) \leq \text{index} < len(primes)$

$-8 \leq \text{index} < 8$

`primes[-9]` → **IndexError: list index out of range**



Basic Python: Unique Indexing in Python

- Printing a list containing integers

```
a = [10,20,30,40,50]

for i in range(len(a)):
    print(a[i])
```

Output:

1	10
2	20
3	30
4	40
5	50

```
a = [10,20,30,40,50]

for i in range(-1, -6, -1):
    print(a[i])
```

Output:

1	50
2	40
3	30
4	20
5	10

	[-5]	[-4]	[-3]	[-2]	[-1]
	[0]	[1]	[2]	[3]	[4]
a	10	20	30	40	50

파이썬 코드로 구현하기: 과제 1


- **step 3)** 최고 기온과 최고 기온이었던 날짜 찾기(스스로 해보기)
- seoul.csv 파일은 사이버캠퍼스 과제에 attach한 파일을 다운받아서 사용

파이썬 코드로 구현하기

- **step 2)** 데이터 중 최고 기온을 실수로 변환하여 한 행씩 출력하기

```
import csv
f = open('seoul.csv')
data = csv.reader(f)
header = next(data)
for row in data :
    row[-1] = float(row[-1])    # 최고 기온을 실수로 변환
    print(row)
f.close()
```

파이썬 코드로 구현하기: 실행결과

```
In [1]: ▶ import csv
max_temp = -999 # 최고 기온 값을 저장할 변수
max_date = '' # 최고 기온이 가장 높았던 날짜를 저장할 변수
f = open('seoul.csv')
data = csv.reader(f)
header = next(data)
for row in data :
    
f.close()
print('기상 관측 이래 서울의 최고 기온이 가장 높았던 날은', max_date + '로, ', max_temp, '도였습니다.')
```

기상 관측 이래 서울의 최고 기온이 가장 높았던 날은 2018-08-01로, 39.6 도였습니다.