

2020년 서울지역의 비가 가장 많이 온 날은 언제였을까?



■문제 해결 방법

- •데이터를 읽어온다.
- ■순차적으로 최고 강수량을 확인한다.
- ■최고 강수량인 날짜의 데이터를 저장한다.
- ■최종 저장된 데이터를 출력한다.

- ■필요한 데이터는?
 - ■날짜, 강수량

	Α	В	С
1	날짜	지점	강수량(mm)
2	2020-01-01	108	0.1
3	2020-01-02	108	
4	2020-01-03	108	
5	2020-01-04	108	
6	2020-01-05	108	
7	2020-01-06	108	1.8
8	2020-01-07	108	46.3
9	2020-01-08	108	11.6
10	2020-01-09	108	
11	2020-01-10	108	
12	2020-01-11	108	
13	2020-01-12	108	
14	2020-01-13	108	
15	2020-01-14	108	
16	2020-01-15	108	
17	2020-01-16	108	
18	2020-01-17	108	
19	2020-01-18	108	
20	2020-01-19	108	0.6
21	2020-01-20	108	

여기서 잠깐!

- 순서 있는 저장 공간 리스트
 - 파이썬 에서는 순서가 있는 데이터를 다룰 때, 리스트 데이터 구조를 사용함
 - ▪대괄호[]로 감싼 부분

```
국민대
고려대
한양대
```

```
for univ in ['국민대','고려대','한양대']:
```

건축학 공간디자인 금속공예 경영학 재무금융회계 경영정보 print(univ)

```
print()
major=['건축학','공간디자인','금속공예','경영학','재무금융회계','경영정보']
for i in major:
 print(i, end=' ')
```

■ 리스트는 전체 데이터를 대괄호로 감싸주고 각각의 값들을 콤마로 구분함

여기서 잠깐!



■ 리스트에 저장된 위치(index)로 값에 접근하기

```
univ = ['국민대','고려대','한양대']
print(univ[0])
print(univ[1])
print(univ[2])
print(univ[-1])
```

국민대 고려대 한양대 한양대



step 1) 데이터 불러서 한 행씩 출력하기

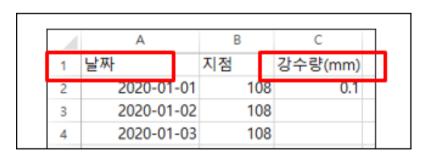
```
import csv
seoul = open('서울강수량2020.csv')
rainfall = csv.reader(seoul)
next(rainfall)
for row in rainfall:
  print(row)
seoul.close()
```

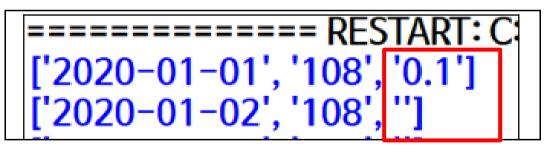
```
['2020-01-01', '108', '0.1']
['2020-01-02', '108', '']
['2020-01-03', '108', '']
['2020-01-04', '108', '']
['2020-01-05', '108', '']
['2020-01-06', '108', '1.8']
['2020-01-07', '108', '46.3']
['2020-01-08', '108', '11.6']
['2020-01-09', '108', '']
['2020-01-10', '108', '']
['2020-01-11', '108', '']
['2020-01-12', '108',
['2020-01-13', '108', '']
['2020-01-14', '108',
['2020-01-15', '108', '']
['2020-01-16', '108', '']
['2020-01-17', '108', ''
```

여기서 잠깐!



- ●현재 강수량 데이터는 숫자 값이 아닌 문자열 이기 때문에, 값을 더하 거나 크기를 비교할 수 없음
- ■숫자로 다루려면 숫자 타입 데이터로 변환을 해주어야 함
- 강수량 데이터는 0.1, 46.3 등의 소수점이 있는 실수이므로 float()함 수를 사용해서 실수 데이터로 변환해야 함
 - -row[2]=float(row[2])







■step 2) 데이터 중 강수량을 실수로 변환하여 한 행씩 출력하기

```
import csv
seoul = open('서울강수량2020.csv')
rainfall = csv.reader(seoul)
                           ['2020-01-01', '108', 0.1]
                           Traceback (most recent call last):
next(rainfall)
                            File "C:/Users/hjjin/OneDrive/바탕 화면/서울 강수량.py", line 10, in (module)
                             row[2]=float(row[2])
for row in rainfall:
                           ValueError: could not convert string to float:
 row[2] = float(row[2])
                                             print(row)
                                             ['2020-01-01', '108', '0.1']
                                             ['2020-01-02', '108', '']
seoul.close()
```



- 문자열 형태의 강수량 데이터를 실수 형태로 바꾸던 중 빈 문자열을 어떤 실수 값으로 바꿔야 할지 몰라서 오류가 발생한 것임
- 빈 값을 대체할 특정 값을 정해 넣어보자.
 - ■최고 강수량으로 나오기 힘든 아주 작은 값을 넣어 이 데이터는 빈 문자 열이 있던 자리야 라고 표현하면 됨

```
if row[2]==":
row[2]=-100 #-100을 넣어 빈 문자열이 있던 자리라고 표시
```

['2020-01-02', '108', '']

```
4. 비가 가장 많이 온 날은 언제였을까?
                                                        ['2020-01-01', '108', 0.1]
                                                        ['2020-01-02', '108', -100.0]
import csv
                                                        ['2020-01-03', '108', -100.0]
                                                        ['2020-01-04', '108', -100.0]
                                                        ['2020-01-05', '108', -100.0]
seoul = open('서울강수량2020.csv')
                                                        ['2020-01-06', '108', 1.8]
                                                        ['2020-01-07', '108', 46.3]
rainfall = csv.reader(seoul)
                                                        ['2020-01-08', '108', 11.6]
                                                        ['2020-01-09', '108', -100.0]
                                                        ['2020-01-10', '108', -100.0]
next(rainfall)
                                                        ['2020-01-11', '108', -100.0]
                                                        ['2020-01-12', '108', -100.0]
for row in rainfall:
                                                        ['2020-01-13', '108', -100.0]
  if row [2] == ":
                                                        ['2020-01-14', '108', -100.0]
    row[2]=-100 #-100을 넣어 빈 문자열이 있던 자리라고 표시
  row[2]=float(row[2])
  print(row)
seoul.close()
```



- ■step 3) 비가 가장 많이 온 날짜 찾기
 - ■최고 강수량을 저장할 변수 max_rain
 - ■가장 비가 많이 온 날짜 저장할 변수 max_date

```
max_rain =0 # 최고 강수량 값을 저장할 변수 
max_date=" # 가장 비가 많이 온 날짜를 저장할 변수
```

```
['2020-01-01', '108', 0.1]

['2020-01-02', '108', -100.0]

['2020-01-03', '108', -100.0]

['2020-01-04', '108', -100.0]

['2020-01-05', '108', -100.0]

['2020-01-06', '108', 1.8]

['2020-01-07', '108', 46.3]

['2020-01-08', '108', 11.6]
```



만약 지금까지의 최고 강수량 보다 현재 행(row)의 최고 강수량이 더 많으면 최고 강수량 날짜 업데이트 최고 강수량 값 업데이트

```
2020년 서울에서 비가 가장 많이 온날은 2020-01-07 이고, 강수량은 46.3 였습니다. _{\mathbf{Y}}
import csv #csv 모듈 불러오기
                                                                                  ''날짜', '지점', '강수량(mm)']
                                                                                  ['2020-01-01', '108', '0.1']
seoul = open('서울강수량2020.csv') #서울강수량2020.csv 파일 읽기 모드로 불러오기
                                                                                  '2020-01-02', '108', '']
                                                                                  <u>''2020-01-03', '108', '']</u>
                                                                                  ''2020-01-04', '108', ''<u>|</u>
rainfall = csv.reader(seoul)
                                                                                  '2020-01-05', '108', '']
                                                                                  ['2020-01-06', '108', '1.8']
next(rainfall)
                                                                                  ['2020-01-07', '108', '46.3']
                                                                                  '2020-01-08', '108', '11.6']
                                                                                  ['2020-01-09', '108', '']
max_rain =0 # 최고 강수량 값을 저장할 변수
                                                                                  <u>'2020-01-10', '108', '']</u>
max_date="# 가장 비가 많이 온 날짜를 저장할 변수
                                                                                  ['2020-01-11', '108', '']
                                                                           ['2020-01-01', '108', 0.1]
for row in rainfall:
                                                                           ['2020-01-02', '108', -100.0]
                  #만약 데이터가 누락되었다면 강수량을 -100으로 저장
 if row [2] == ":
   row[2] = -100
                                                                           ['2020-01-03', '108', -100.0]
                                                                           ['2020-01-04', '108', -100.0]
 row[2]=float(row[2]) #문자열로 저장된 강수량 값을 실수로 변환
                                                                           ['2020-01-05', '108', -100.0]
                                                                           ['2020-01-06', '108', 1.8]
 if max_rain (row[2]: #만약 지금까지 최고 강수량보다 더 많다면 업데이트
   max_date=row[0]
                                                                           ['2020-01-07', '108', 46.3]
   max_rain =row[-1]
                                                                           ['2020-01-08', '108', 11.6]
```

print('2020년 서울에서 비가 가장 많이 온날은',max_date,'이고, 강수량은',max_rain,'였습니다.') #파일 출력 seoul.close() #파일 닫기