

오늘은 Chapter 8들이시면 되겠습니다. 아마 강의 내용이 뭔가 많을거예요. 갑자기 어렵게 느껴질 수도 있어요. 그래도 파이팅.

오늘은 단어 찾기 프로그램을 만들기에 앞서 파일 입출력과 dict타입에 익숙해지는 시간을 가질까 해요. 파일 입출력은 다음주에 배우는 내용이지만, 미리미리 안다고 나쁠 것은 없으니 오늘 조금만 나가봐요. Java, C를 하면서 파일 입출력은 다들 공부하셨을 거예요. 파이썬도 크게 다른 건 없어요. 다들 구글에서 파이썬 파일 입출력에 대해 검색해보고 각자 공부해봐요.

‘KakaoTalk_20210112_0314_25_973_group.txt’ 파일이 있을거예요. 11일 저녁까지 저희 카톡방에서 했던 대화들이 담겨있습니다. 이 카톡파일을 분석해볼거예요.

1. 대화를 가장 많이 한 사람
 2. 이모티콘을 가장 많이 보낸 사람
 3. 사진을 가장 많이 보낸 사람
 4. 대화 삭제를 가장 많이 한 사람
 5. 날짜별 대화량
 6. 시간대별 대화량
 7. 가장 많이 나온 단어

위 6가지에 대해서 분석해볼게요. kakaotalk.py 파일이 있을거예요. 파일 입출력과 dict 초기화는 코드 작성이 되어있어요. txt파일과 chat리스트를 잘 살펴보고 아래 문제들을 풀어보세요.

txt파일을 보면 아시겠지만, txt파일 첫부분과 초대되었다는 메시지를 제외하고는, 모든 줄의 시작이 ‘=’ 또는 ‘[’예요. 반대로, ‘=’나 ‘[’로 시작 하지 않는 줄은 그럼 무시하고 넘어가도 되겠네요. ‘=’인 경우는 해당 줄에 날짜가 나오고, ‘[’인 경우는 해당 줄에 대화가 나와요.

dict타입을 value값을 기준으로 정렬해야 해요. 다음 코드를 이용해보세요.

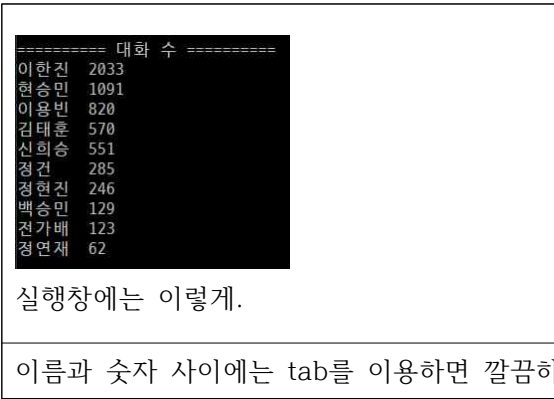
```
dict(sorted(x.items(), key=lambda item: item[1]))
```

위 코드에서 x가 dict타입입니다. x대신에 해당 dict의 이름을 넣으면 되겠죠?

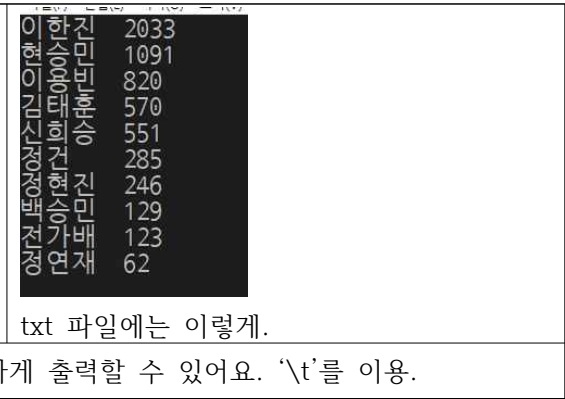
1. 대화를 가장 많이 한 사람.

conversations_number는 각 멤버를 key로, 대화한 수를 value값으로 가지는 딕셔너리예요. txt 파일에는 한 줄에 한명씩 톡한게 들어있어요. 이걸 이용하고, 마지막에는 출력과 파일 출력도 같이 해봐요. write()함수를 이용할 수 있어요.

conversations_number.txt파일을 만들어서 내림차순으로 출력해봅시다.



실행창에는 이렇게.



txt 파일에는 이렇게.

이름과 숫자 사이에는 tab를 이용하면 깔끔하게 출력할 수 있어요. ‘\t’를 이용.

2. 이모티콘을 가장 많이 보낸 사람

마찬가지예요. `emoticon_frequency`도 각 멤버를 `key`로 이모티콘 사용 횟수를 `value`로 가져요. 이모티콘을 보낸 경우, `txt`파일에는 다음과 같이 표시되어 있어요.

A screenshot of a chat log showing a message from '이용빈' at '오후 7:57' with the text '이모티콘'.

이를 이용해 `emoticon_frequency.txt`를 만들고 1번해서 했던 것과 똑같이 출력해보세요.

3. 사진을 가장 많이 보낸 사람

`pictures_frequency`도 각 멤버를 `key`로, 사진 사용 횟수를 `value`로 가져요.

A screenshot of a chat log showing a message from '이한진' at '오후 8:08' with the text '사진'.

역시 `pictures_frequency.txt`를 만들고 순서대로 출력하세요.

4. 대화 삭제를 가장 많이 한 사람

`delete_frequency`를 이용하시면 되고 이제 설명 안해도 아시죠?

A screenshot of a chat log showing a message from '정현진' at '오전 10:49' with the text '삭제된 메시지입니다.'.

5. 날짜별 대화량

`date_conversations`는 조금 방식이 달라요. 이 딕셔너리는 비어있어요. 날짜가 나올 때마다 날짜를 `key` 값에 '@@@@년-@@월-@@일-@@요일'과 같은 문자열로 추가하고, 해당 날짜에 대화한 빈도수를 `value`값에 넣으시면 되요. 이 dict타입은 정렬할 필요 없이 날짜순으로 출력하면 됩니다.

날짜가 바뀔때마다 '----- 2020년 12월 22일 화요일 -----' 이런 줄이 나오고, 다음에 저런 줄이 나오기 전까지의 모든 대화는 12월 22일에 한 대화라는 것을 생각해보시면 조금 쉬워요.

6. 시간대별 대화량

'-'로 시작하는 줄이 아닌 모든 '['로 시작하는 줄에는 아래 사진과 같이 이름 날짜 내용 순으로 나와요.

ex1) [정건] [오전 11:01] 형 저 어제

ex2) [정현진] [오후 2:40] 저거 한입컷 가능할거같은데

`time_conversations`는 `key`값이 0~23이고, `value`값이 해당 시간에 대화의 개수예요.

'14:40'이 아니라 '오후 2:40'으로 표기되는 것에 주의하시고 ex1같은 경우에는 `time_conversations[11]`에 1을 추가하고, ex2같은 경우에는 `time_conversations[14]`에 1을 추가하면 되겠죠?

이것도 정렬할 필요 없이 출력하시면 되요.

7. 가장 많이 나온 단어

모든 대화는 '[이름] [시간정보] 대화' 순으로 진행되요. 대화에서 단어는, 무조건 띄어쓰기를 기준으로 한 단어라고 칠게요. `word_frequency`도 5번과 같이 비어있는 dict예요. `key`값을 단어로, `value`값을 해당 단어가 나온 빈도수로 설정할게요.

[현승민] [오후 6:32] 저문제는 구구콘이나 구구크러스트 먹으며 풀어야 제맛

위 경우에는 단어가 '저문제는', '구구콘이나', '구구크러스트', '먹으며', '풀어야', '제맛' 이렇게 나뉘고, `word_frequency['저문제는'] += 1`과 같이 코드 짜시면 되요. 단어는 워낙 많다보니, 실행창에는 출력할 필요 없이 `txt` 파일만 만드시면 되요.

출력 예시는 스크린샷을 같이 올려놨으니 보고 해당 양식에 맞게 출력해보세요.