

**[수업 목표]**

1. 파이썬으로 텍스트 파일을 읽고 쓸 수 있다.
2. 문자열에서 원하는 부분만 잘라낼 수 있다.
3. 파이썬으로 워드클라우드를 만들 수 있다.

01. 오늘 배울 것
02. 텍스트파일 읽고 쓰기
03. 워드클라우드 바로가기
04. 데이터 클렌징
05. 워드 클라우드
06. 끝 & 숙제 설명

- 오늘은 드디어 카톡 내용으로 워드 클라우드를 만들어보겠습니다!

▼ 이번 시간이 끝나면 완성하는 것



### ▼ 1) 파이썬으로 텍스트 파일 쓰기

- 파이썬으로 텍스트 파일을 작성할 때는 내장 함수인 `open()` 을 이용합니다. 메일에 파일첨부할 때도 봤었죠?
- 우선 아래 코드를 새 파이썬 파일에 붙여넣고 실행해보겠습니다.

```
f = open("test.txt", "w", encoding="utf-8")
f.write("안녕, 스파르타!")
f.close()
```

- 첫 번째 변수는 파일명, 두 번째 변수는 쓰기 모드, 세 번째 변수는 한글을 지원하는 인코딩을 뜻합니다.

**▲ 쓰기 모드에서는 파일이 존재하면 기존 내용을 덮어씌워버리니 조심하세요!**

- 파이썬에서 연 파일은 꼭 잘 닫아주어야하는데, `open()` 과 `.close()` 사이에 코드가 길어지면 잊어버리기 쉽습니다. 그래서 위의 코드를 더 깔끔하게 아래처럼 적는 것이 선호됩니다. `with` 를 이용하면 블록 안의 코드를 다 실행하고 자동으로 닫아줍니다.

```
with open("test.txt", "w", encoding="utf-8") as f:
    f.write("안녕, 스파르타!")
```

- 여러 줄로 적고 싶을 때는 사이에 줄바꿈 이스케이프 코드 `\n` 을 넣어주면 됩니다. 아래처럼 반복문에서 응용하면 좋겠죠?

```
with open("test.txt", "w", encoding="utf-8") as f:
    f.write("각 줄마다 번호를 적은 파일입니다.\n")
    for i in [1,2,3,4,5]:
        f.write(f"이것은 {i}번째 줄입니다.\n")
```

## ▼ 2) 텍스트파일 읽기

- 이번에는 파일을 읽어보겠습니다! 읽기모드(`r`)를 이용하면 되겠죠?

### ▼ [코드스니펫] - 텍스트 파일 읽기

```
with open("test.txt", "r", encoding="utf-8") as f:
    lines = f.readlines()
    for line in lines:
        print(line)
```

- `print` 할 때마다 새 줄에 출력하고, `line` 끝에 `\n` 이 있기 때문에 각 줄 사이에 엔터를 2번 친 것 같은 효과가 나는데요! (괜찮아요~!)

## 03. 워드클라우드 바로가기

### ▼ 4) 카카오톡 대화내용 준비하기

☞ PC 카톡에서 채팅방을 연 후 메뉴 > 대화 내용 > 대화내용내기 해서 대화 내용을 텍스트 파일로 저장한 후, sparta 폴더로 옮겨주세요. 또는, 아래 파일을 이용하셔도 좋습니다.

### ▼ [코드스니펫] - 채팅 내용 예시

```
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/68ea11c5-5fbd-40f8-9390-7fd7eacb37f3/kakaotalk.zip
```

### ▼ 5) 카카오톡 텍스트파일 읽기

- 텍스트 파일의 각 줄을 읽어 새 문자열인 `text` 에 저장하는 방법을 사용하겠습니다.

```
text = ""
# 파일 이름은 맞게 바꿔주세요!
with open("kakaotalk.txt", "r", encoding="utf-8") as f:
    lines = f.readlines()
    for line in lines:
        text+=line

print(text)
```

### ▼ 6) 만들 준비 하기

- 패키지 설치

`wordcloud` 라는 패키지를 이용하여 워드 클라우드를 만들어보도록 하겠습니다. Settings/Preferences > Project Interpreter에서 설치해주세요.

- 한글 폰트 찾기

파이썬에서 그림을 그릴 때 기본으로 사용하는 폰트가 한글을 지원하지 않는 경우가 많습니다. 그래서 한글을 지원하는 폰트로 바꾸어주려면 우리가 직접 폰트 파일이 저장되어있는 경로를 찾아 넣어주어야해요.

1. 사용 가능한 폰트 목록 보기

컴퓨터에 있는 폰트는 엄청 많으니까, 고딕 폰트만 출력해보도록 하겠습니다.

▼ [코드스니펫] - 폰트 검색

```
import matplotlib.font_manager as fm

# 이용 가능한 폰트 중 '고딕'만 선별
for font in fm.fontManager.ttflist:
    if 'Gothic' in font.name:
        print(font.name, font.fname)
```



`matplotlib`은 그래프 그릴 때 쓰는 패키지로, 실제로 `wordcloud`가 이미지를 만드는 데 쓰기 때문에 자동으로 같이 설치된답니다.

2. 한글 폰트 고르기

한글을 지원하는 폰트로는 '**AppleGothic**', '**Malgun Gothic**', '**NanumGothic**' 등이 있습니다. 위의 출력된 목록에서 찾아 경로를 복사해주세요.

```
font_path = 'C:\Windows\Fonts\malgunbd.ttf'
```

3. (Windows만) 특수문자 바꾸주기

윈도우즈 운영체제에서 쓰는 컴퓨터에서는 경로에 백슬래시(\\)를 사용하는데, 파이썬 문자열에서는 특수한 기능(이스케이프 코드라고 합니다)으로 쓰이기 때문에 처리를 해주어야합니다. 포워드 슬래시(/)로 바꿔주세요!

```
font_path = 'C:/Windows/Fonts/malgunbd.ttf'
```

▼ 7) 워드 클라우드 만들기



코드 자체는 엄청 간단해요! 이게 바로 파이썬 패키지의 마법~!

▼ [코드스니펫] - 워드 클라우드 만들기

```
from wordcloud import WordCloud

wc = WordCloud(font_path=font_path, background_color="white", width=600, height=400)
wc.generate(text)
wc.to_file("result.png")
```

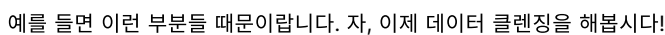
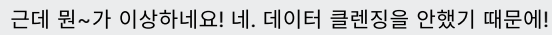
▼ 전체 코드

```
from wordcloud import WordCloud

text = ""
# 파일 이름은 맞게 바꿔주세요!
with open("kakaotalk.txt", "r", encoding="utf-8") as f:
    lines = f.readlines()
    for line in lines:
        text += line

font_path = 'C:/Windows/Fonts/CoreGTM3.otf'

wc = WordCloud(font_path=font_path, background_color="white", width=600, height=400)
wc.generate(text)
wc.to_file("result.png")
```



## 04. 데이터 클렌징

- 전처리(preprocessing) 과정의 하나로, 자료에서 불완전하거나, 비정확하거나, 관련 없는 부분을 찾아 삭제하거나 수정하는 것을 말합니다. 테이블에 빈 칸이 있거나, 단위가 잘못되었거나, 오타가 있거나 등등의 경우에 원하는 결과를 얻을 수 없기 때문에 분석하기 전에 미리 처리를 해주는 것이죠. 우리의 경우에는 내보내기한 카카오톡을 열어보면 날짜, 시간, 프로필 이름 등의 무의미한 부분이 반복적으로 나타나기 때문에 이것들을 지워주어야 더 흥미로운, 실제로 사람들이 사용한 단어를 강조할 수 있습니다.

1. 워드클라우드 그리는 부분은 잠시 주석처리 해둘까요? 여기서 시작!

스파르타온라인 파이썬 혼자놀이 패키지 - 3일차

2. 우선 파일의 처음 두 줄은 특방 이름과 저장한 날짜이기 때문에 없애줘도 되겠죠? 슬라이싱을 사용하여 반복문 자체를 세 번째 줄부터 들어가도록 할 수 있습니다!

```
for line in lines[2:]:
    text+=line
```

3. 이번엔 들어오셨습니다 나가셨습니다 시스템 메시지를 제거하고, 대화만 남게 해볼게요!

👉 생각: 각 줄에는 `]` `[` 가 포함되어 있는데요! 이 것을 이용하기!

```
for line in lines:
    if ']' [' in line:
        text += line
```

4. 거기에 이렇게 하면, 이름, 시간을 제외하고 대화 내용만 가져올 수 있겠죠!

👉 생각: 만약  로 나누게 되면 어떨까요?

```
for line in lines:
    if ']' [' in line:
        text += line.split(' ')[2]
```

5. 얼추 깔끔한 결과가 나온 것 같죠?

아 근데 머라고 불리 짚짜  
ㅋㅋㅋㅋ  
안줄려(요?)  
머라고 불리야되냐그  
사람ㅁ  
ㅋㅋㅋ  
ㅋㅋㅋㅋㅋㅋ

- ## 6. 주석을 풀고 워드클라우드를 그려봅시다!

👉 아~ 아직도 뭔가 부족한데요!

네! 그렇습니다. 이모티콘, 사진,  $\pi$ ,  $\tau$  등의 문자열들을 제거해주면 좋겠죠!



## ▼ 10) 플랫폼에 특화된 데이터클렌징

## 1. 이모티콘, 사진, \, \n, \t 등의 문자열들 제거하기

python의 replace를 이용하면 편리해요!  
어렵진 않는데 오타가 발생할 수 있으니 이것은 코드스니펫으로!

### ▼ [코드스니펫] - 플랫폼에 특화된 데이터렌징

```
for line in lines:
    if ']' [' in line:
        text += line.split(' ')[2].replace(';', '').replace('\'', '').replace('\"', '').replace('사진\n', '').replace(''
```

## 2. 결과를 보면, 이제 꽤 그럴싸 하네요!

### ▼ 완성코드

```
from wordcloud import WordCloud

text = ""
# 파일 이름은 맞게 바꿔주세요!
with open("kakaotalk.txt", "r", encoding="utf-8") as file:
    lines = file.readlines()
    for line in lines:
        if ']' [' in line:
            text += line.split(' ')[2].replace(';', '').replace('\'', '').replace('\"', '').replace('사진\n', '').replace('

font_path = 'C:/Windows/Fonts/CoreGTM3.otf'

wc = WordCloud(font_path=font_path, background_color="white", width=600, height=400)
wc.generate(text)
wc.to_file("result.png")
```



## 05. 워드 클라우드

### ▼ 11) 원하는 모양으로 만들기

- 직사각형 모양 뿐만 아니라, 원하는 어떤 모양이든 흰색 배경의 이미지가 있으면 만들 수 있습니다. 예시로 간단하게 구름 모양의 워드 클라우드를 만들어볼게요.

### ▼ [코드스니펫] - 구름 이미지 다운로드

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/28e6c56f-c254-47c5-944a-31c7fec264d1/cloud.png>



- 이 이미지를 스텐실(컴퓨팅에서는 mask 🎨 라는 표현을 사용합니다)로 사용해서 검은 부분에만 글씨를 채워넣도록 하겠습니다.

#### ▼ [코드스니펫] - 마스킹된 워드 클라우드 만들기

```
from PIL import Image
import numpy as np

mask = np.array(Image.open('cloud.png'))
wc = WordCloud(font_path=font_path, background_color="white", mask=mask)
wc.generate(text)
wc.to_file("result_masked.png")
```



## 06. 끝 & 속제 설명

💬 워드 클라우드 만들어서 친구에게 자랑하기! 😊

- 원하는 카카오톡 채팅방 내용을 이용해서 원하는 모양으로 워드 클라우드를 만들어보시다!

👉 그 채팅방에 올려 자랑해보기! "이거 내가 한거야!" 😎

▼ 이렇게 되면 완성

