2022 LSK-시몬느 언어학학교 - 언어연구를 위한 Python 프로그래밍 (초급)

03 Pandas Library

윤태진 교수
성신여자대학교 영어영문학과

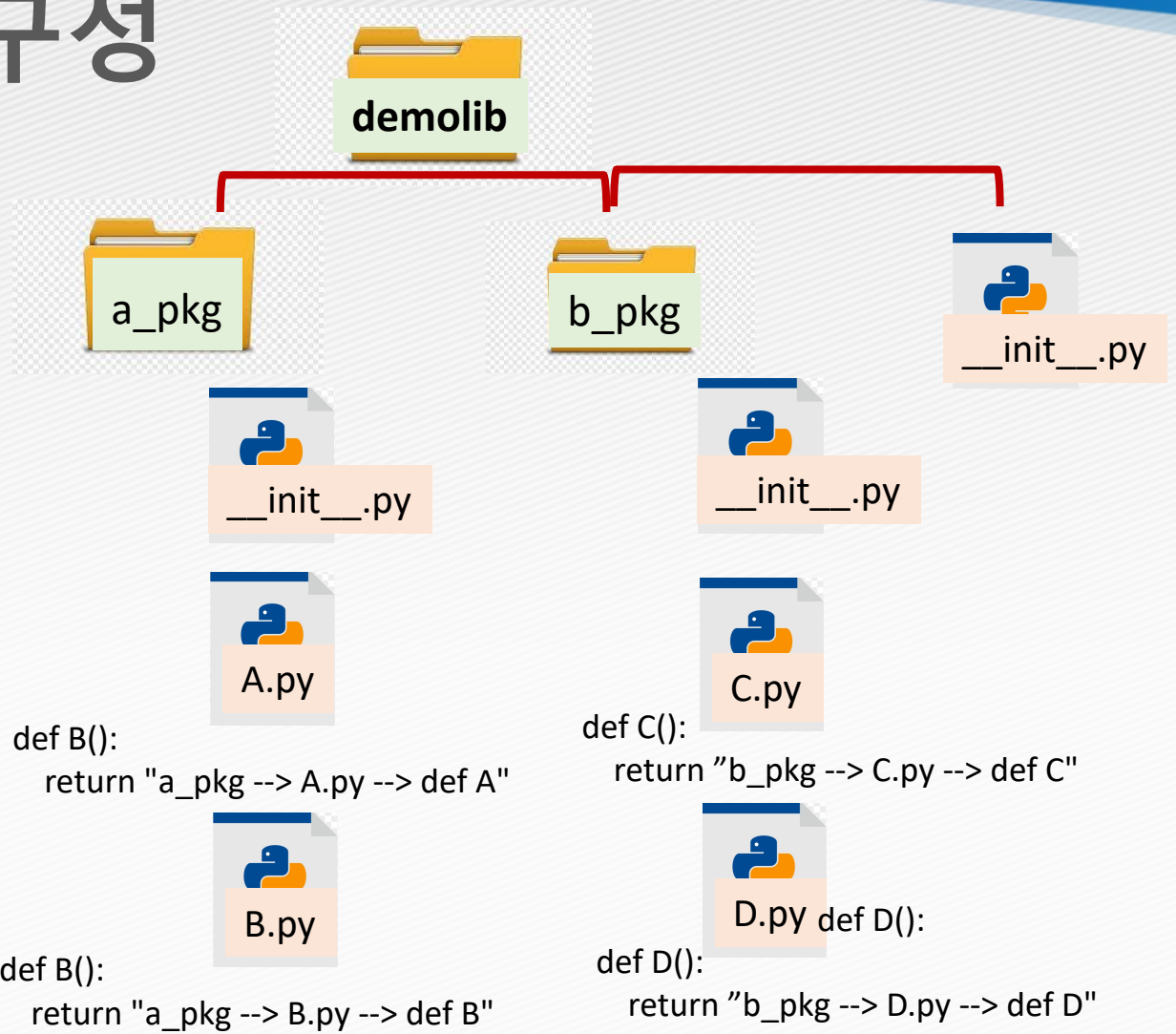
강의 내용

1. 패키지 사용하기
2. Pandas 설치 및 사용법
3. Pandas-based Text Processing
4. Pandas-based Text Processing

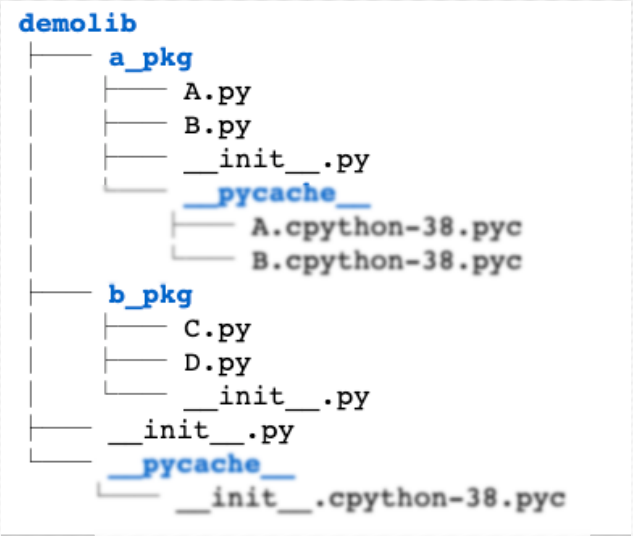
01

Package 사용하기

라이브러리 구성



- 폴더(디렉터리) 안에 __init__.py 파일이 있으면 해당 폴더는 패키지로 인식됩니다.
- 기본적으로 __init__.py 파일의 내용은 비워 둘 수 있습니다
  - 파이썬 3.3 이상부터는 __init__.py 파일이 없어도 패키지로 인식됨.
  - 하지만 하위 버전에도 호환되도록 __init__.py 파일을 작성하는 것을 권장



import 패키지.모듈

패키지.모듈.변수

패키지.모듈.함수()

패키지.모듈.클래스()

from 패키지.모듈 import 변수

from 패키지.모듈 import 함수

from 패키지.모듈 import 클래스

함수()

클래스()

current_working_python_file

```
demolib
├── a_pkg
│   ├── A.py
│   ├── B.py
│   ├── __init__.py
│   └── __pycache__
│       ├── A.cpython-38.pyc
│       └── B.cpython-38.pyc
├── b_pkg
│   ├── C.py
│   ├── D.py
│   └── __init__.py
├── __init__.py
└── __pycache__
    └── __init__.cpython-38.pyc
```

```
from demolib import a_pkg
a_pkg.A.A()
```

```
'a_pkg --> A.py --> def A'
```

```
from demolib.a_pkg import A, B
```

```
print(A.A())
print(B.B())
```

```
a_pkg --> A.py --> def A
a_pkg --> B.py --> def B
```

01

pandas 설치 및 사용법

Pandas: Excel on Steroid

Pandas의 주요 특징
- 자유로운 데이터 변환
- 엑셀처럼 활용도 높은 DataFrame 객체
- 데이터 구조에 대한 변환

Jupyter Notebook에서의 설치

! pip3 install pandas
! python3 –m pip install pandas

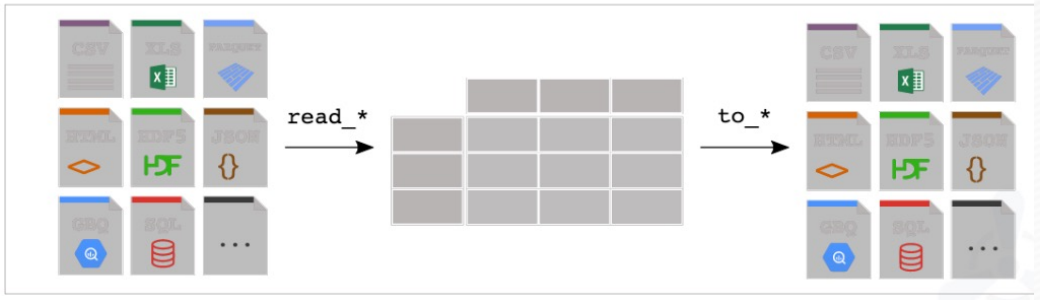
https://www.kdnuggets.com/2020/11/pandas-steroids-dask-python-data-science.html

파일의 경로 설정

```
import pandas as pd
```

파일 불러오기

- 주피터 노트북과 같은 위치에 csv 파일을 옮기기
- Pandas의 read_* 기능을 통해 파일 읽어오기
  - 불러온 파일은 DataFrame이라는 행렬 형태



cvs 파일을 Pandas로 불러오기

```
df = pd.read_csv("data/iris.csv")
df.head()
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa

기상청

CSV파일을 주피터 노트북으로 불러오기

pd.read_csv("bd_kma_weather.csv")

예시

```
df.head()
```

```
df = pd.read_csv("bd_kma_weather.csv")
```

info() 메소드로 데이터의 요약 보기

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal.length  150 non-null    float64
 1   sepal.width   150 non-null    float64
 2   petal.length  150 non-null    float64
 3   petal.width   150 non-null    float64
 4   variety       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

범주형 자료

```
# 컬럼명만 출력해 봅니다.
df.columns
```

```
Index(['sepal.length', 'sepal.width', 'petal.length', 'petal.width',
       'variety'],
      dtype='object')
```

```
# 데이터 타입만 출력합니다.
df.dtypes
```

```
sepal.length    float64
sepal.width     float64
petal.length    float64
petal.width     float64
variety          object
dtype: object
```

```
# 지점명 컬럼을 불러옵니다.
df["sepal.length"].describe()
```

```
count    150.000000
mean       5.843333
std        0.828066
min        4.300000
25%        5.100000
50%        5.800000
75%        6.400000
max        7.900000
Name: sepal.length, dtype: float64
```

```
df["sepal.length"].mean()
```

```
5.843333333333334
```

범주형 column의 범주별 데이터 수 count하기

```
df['variety'].value_counts()
```

```
Setosa          50
Versicolor      50
Virginica       50
Name: variety, dtype: int64
```

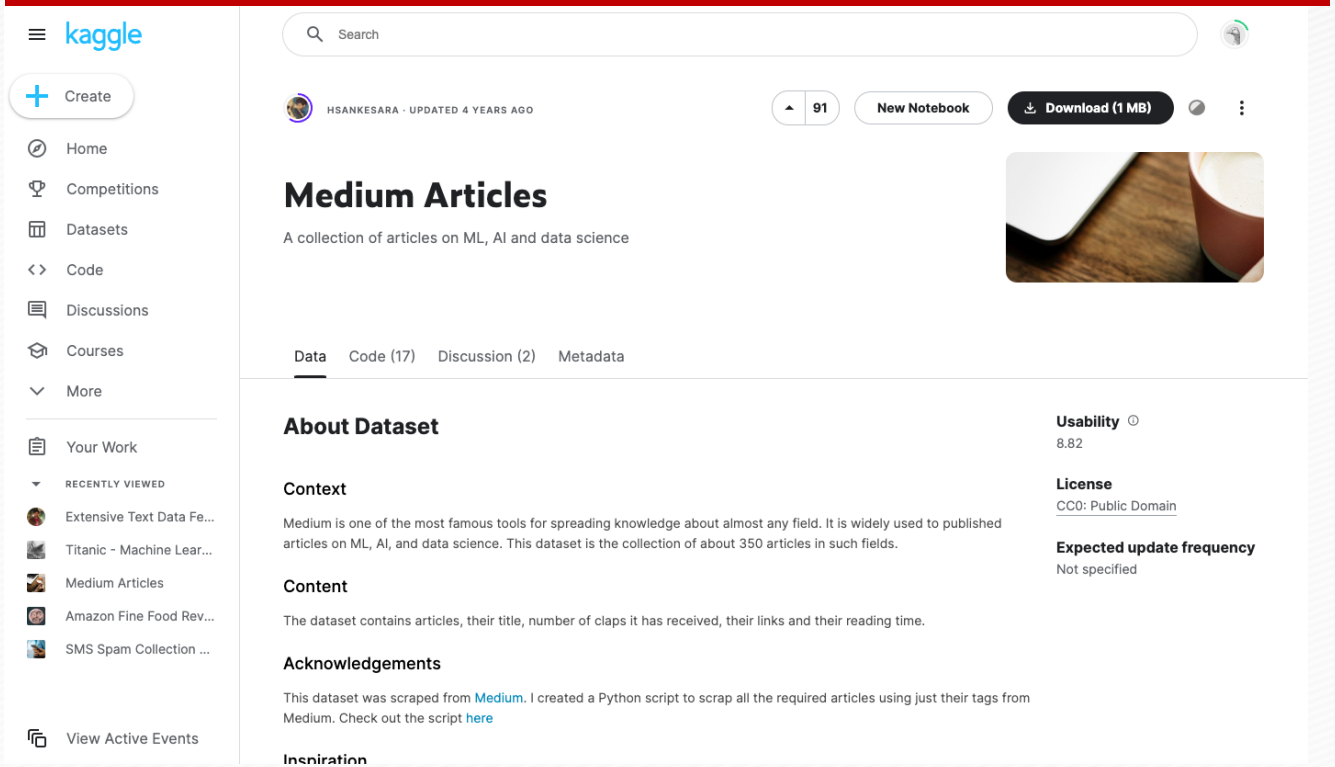
01

Pandas-based Text Processing

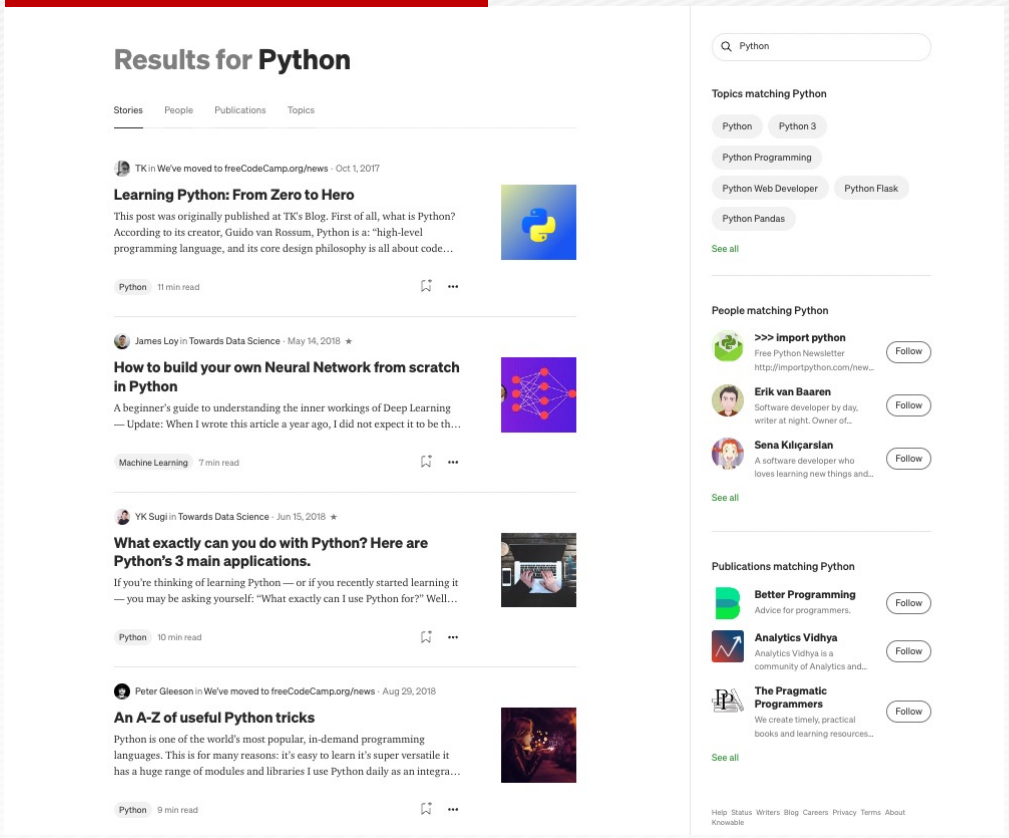
```
import re
import pandas as pd
```

```
df = pd.read_csv("data/articles.csv")
print(df.shape)
print(df.columns)
df.head()
```

https://www.kaggle.com/datasets/hsankesara/medium-articles



https://medium.com/



```
df.sample()
```

	author	claps	reading_time	link	title	text
25	Netflix Technology Blog	365	10	https://medium.com/netflix-techblog/netflix-re...	Netflix Recommendations: Beyond the 5 stars (P...	by Xavier Amatriain and Justin Basilico (Perso...

```
df['title'][25]
```

```
'Netflix Recommendations: Beyond the 5 stars (Part 2)'
```

```
df['text'][25][:100]
```

```
'by Xavier Amatriain and Justin Basilico (Personalization Science and Engineerin
g)\nIn part one of thi'
```

```
print(f"Number of samples: {df.shape[0]}\n")
```

```
Number of samples: 337
```

In [29]:
```
print(f"Sample exceprts:\n")
print(f"Excerpt - 1: {df.iloc[0]['title']} \n{df.iloc[0]['text']}\n")
```

```
Sample exceprts:

Excerpt - 1: Chatbots were the next big thing: what happened? — The Startup — Me
dium
Oh, how the headlines blared:
Chatbots were The Next Big Thing.
Our hopes were sky high. Bright-eyed and bushy-tailed, the industry was ripe for
a new era of innovation: it was time to start socializing with machines.
And why wouldn't they be? All the road signs pointed towards insane success.
At the Mobile World Congress 2017, chatbots were the main headliners. The confer
ence organizers cited an 'overwhelming acceptance at the event of the inevitable
shift of focus for brands and corporates to chatbots'.
In fact, the only significant question around chatbots was who would monopolize
the field, not whether chatbots would take off in the first place:
One year on, we have an answer to that question.
No.
```

```
text = """That’s why it’s still impossible to imagine effective customer support,
sales or marketing without the essential human touch: empathy and emotional
intelligence."""
print("List of words:", str(text).split())
print("Set of words:", set(str(text).split()))
print("Word Count:", len(set(str(text).split())))
```

```
List of words: ['That’s', 'why', 'it’s', 'still', 'impossible', 'to', 'imagine',
'effective', 'customer', 'support,', 'sales', 'or', 'marketing', 'without', 'th
e', 'essential', 'human', 'touch:', 'empathy', 'and', 'emotional', 'intelligenc
e.']
Set of words: {'why', 'or', 'intelligence.', 'support,', 'sales', 'essential',
'emotional', 'without', 'effective', 'empathy', 'impossible', 'to', 'human', 'im
agine', 'touch:', 'marketing', 'the', 'That’s', 'it’s', 'and', 'customer', 'stil
l'}
Word Count: 22
```

```
df['text_num_words'] = df["text"].apply(lambda x: len(set(str(x).split())))
```

```
df.head(1)
```

	author	claps	reading_time	link	title	text	text_num_words
0	Justin Lee	8.3K	11	https://medium.com/swlh/chatbots-were-the-next...	Chatbots were the next big thing: what happene...	Oh, how the headlines blared:\nChatbots were T...	1031

```
text = """That’s why it’s still impossible to imagine effective customer support, sales or marketing without the essential human touch: empathy and emotional intelligence."""
```

```
print("str:", str(text))
print("length:", len(str(text)))
```

```
str: That’s why it’s still impossible to imagine effective customer support, sales or marketing without the essential human touch: empathy and emotional intelligence.
length: 162
```

```
df["text_num_chars"] = df["text"].apply(lambda x: len(str(x)))
```

```
df.head(1)
```

	author	claps	reading_time	link	title	text	text_num_words
0	Justin Lee	8.3K	11	https://medium.com/swlh/chatbots-were-the-next...	Chatbots were the next big thing: what happene...	Oh, how the headlines blared:\nChatbots were T...	1031

```
df['text_num_paragraphs'] = df['text'].apply(lambda x: len(x.split("\n")))
```

```
df.head(1)
```

	author	claps	reading_time	link	title	text	text_num_words
0	Justin Lee	8.3K	11	https://medium.com/swlh/chatbots-were-the-next...	Chatbots were the next big thing: what happene...	Oh, how the headlines blared:\nChatbots were T...	1031

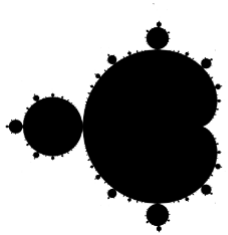
```
df['text_num_sent'] = df['text'].apply(lambda x: len(str(x).split('.')))
```

```
df.head(1)
```

	author	claps	reading_time	link	title	text	text_num_words
0	Justin Lee	8.3K	11	https://medium.com/swlh/chatbots-were-the-next...	Chatbots were the next big thing: what happene...	Oh, how the headlines blared:\nChatbots were T...	1031

```
import numpy as np
df['text_mean_word_len'] = df['text'].apply(lambda x:
                                            np.mean([len(w) for w in str(x).split()]))
df.head()
```

	author	claps	reading_time	link	title	text	text_num_words
0	Justin Lee	8.3K	11	https://medium.com/swlh/chatbots-were-the-next...	Chatbots were the next big thing: what happene...	Oh, how the headlines blared:\nChatbots were T...	1031
1	Conor Dewey	1.4K	7	https://towardsdatascience.com/python-for-data...	Python for Data Science: 8 Concepts You May Ha...	If you've ever found yourself looking up the s...	637
2	William Koehrsen	2.8K	11	https://towardsdatascience.com/automated-featu...	Automated Feature Engineering in Python – Towa...	Machine learning is increasingly moving from h...	764
3	Gant Laborde	1.3K	7	https://medium.freecodecamp.org/machine-learni...	Machine Learning: how to go from Zero to Hero ...	If your understanding of A.I. and Machine Lear...	677



TextBlob

Star 8,226

TextBlob is a Python (2 and 3) library for processing textual data. It provides a consistent API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more.

Useful Links

TextBlob @ PyPI
TextBlob @ GitHub
Issue Tracker

Stay Informed

Follow @sloria

Donate

If you find TextBlob useful, please consider supporting its author:

TextBlob: Simplified Text Processing

Release v0.16.0. (Changelog)

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

```
from textblob import TextBlob

text = '''
The titular threat of The Blob has always struck me as the ultimate movie
monster: an insatiably hungry, amoeba-like mass able to penetrate
virtually any safeguard, capable of--as a doomed doctor chillingly
describes it--"assimilating flesh on contact.
Snide comparisons to gelatin be damned, it's a concept with the most
devastating of potential consequences, not unlike the grey goo scenario
proposed by technological theorists fearful of
artificial intelligence run rampant.
'''

blob = TextBlob(text)
blob.tags           # [('The', 'DT'), ('titular', 'JJ'),
                    #  ('threat', 'NN'), ('of', 'IN'), ...]

blob.noun_phrases   # WordList(['titular threat', 'blob',
                    #           'ultimate movie monster',
                    #           'amoeba-like mass', ...])

for sentence in blob.sentences:
    print(sentence.sentiment.polarity)
# 0.060
# -0.341
```

```
text = ' '.join(str(x) for x in df['text'])
print(text[:100])
```

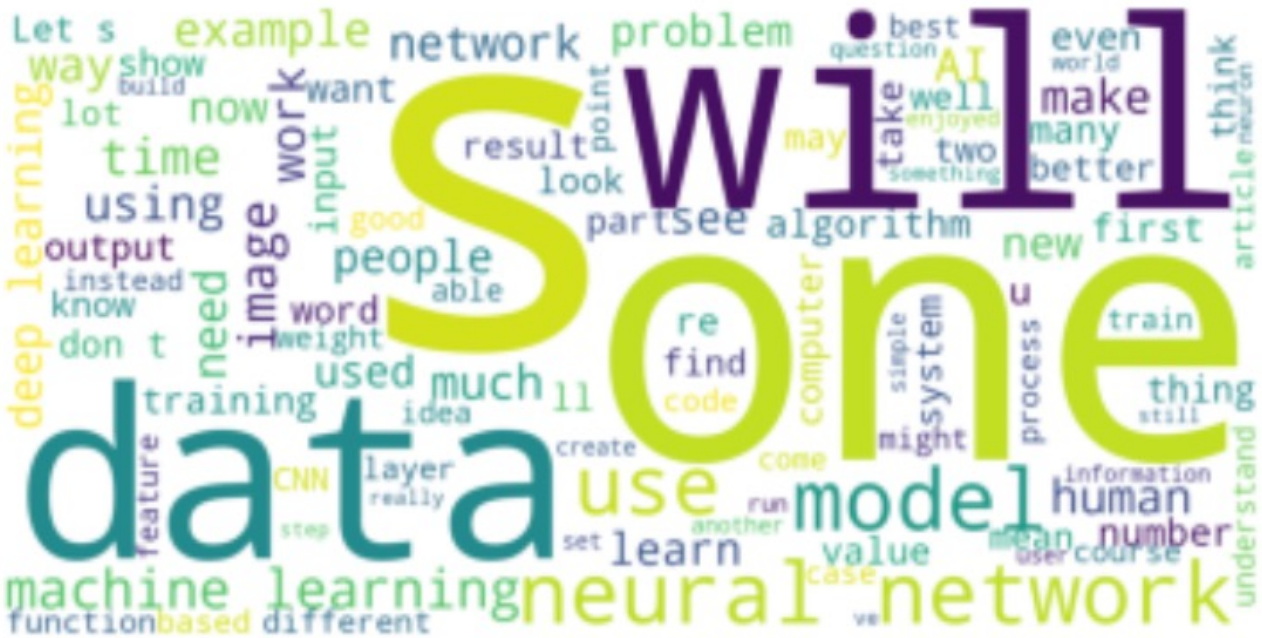
```
Oh, how the headlines blared:
Chatbots were The Next Big Thing.
Our hopes were sky high. Bright-eyed
```

```
import wordcloud as wc
import matplotlib.pylab as plt
```

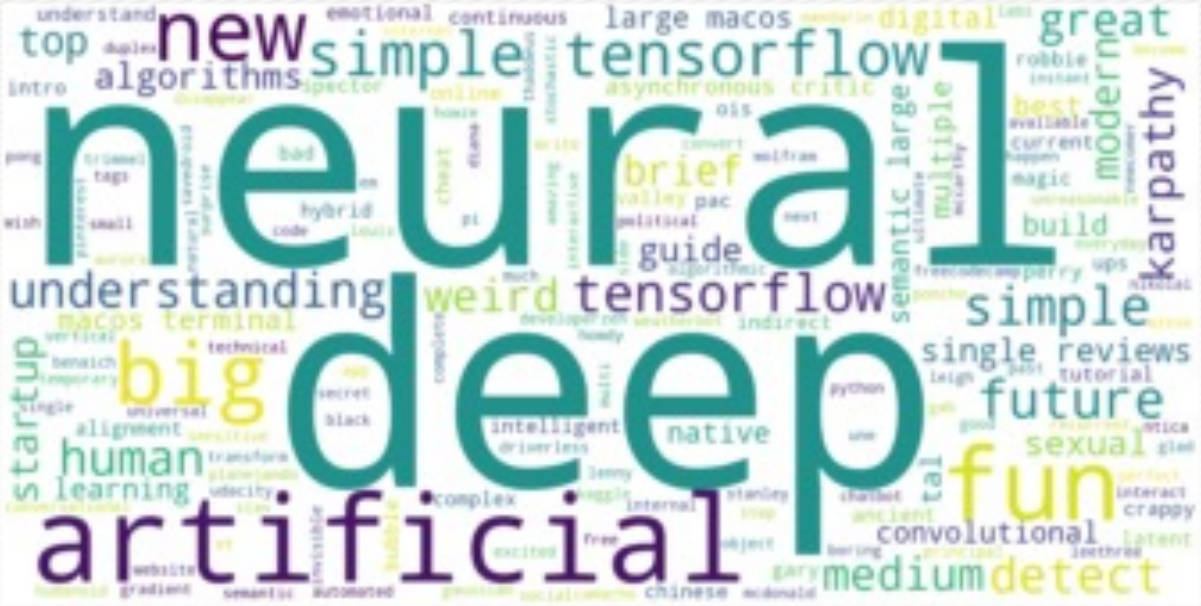
```
wordcloud = wc.WordCloud(width=1600, height=800, max_words=100, background_color="white").gener
plt.imshow(wordcloud)
plt.axis("off")
```

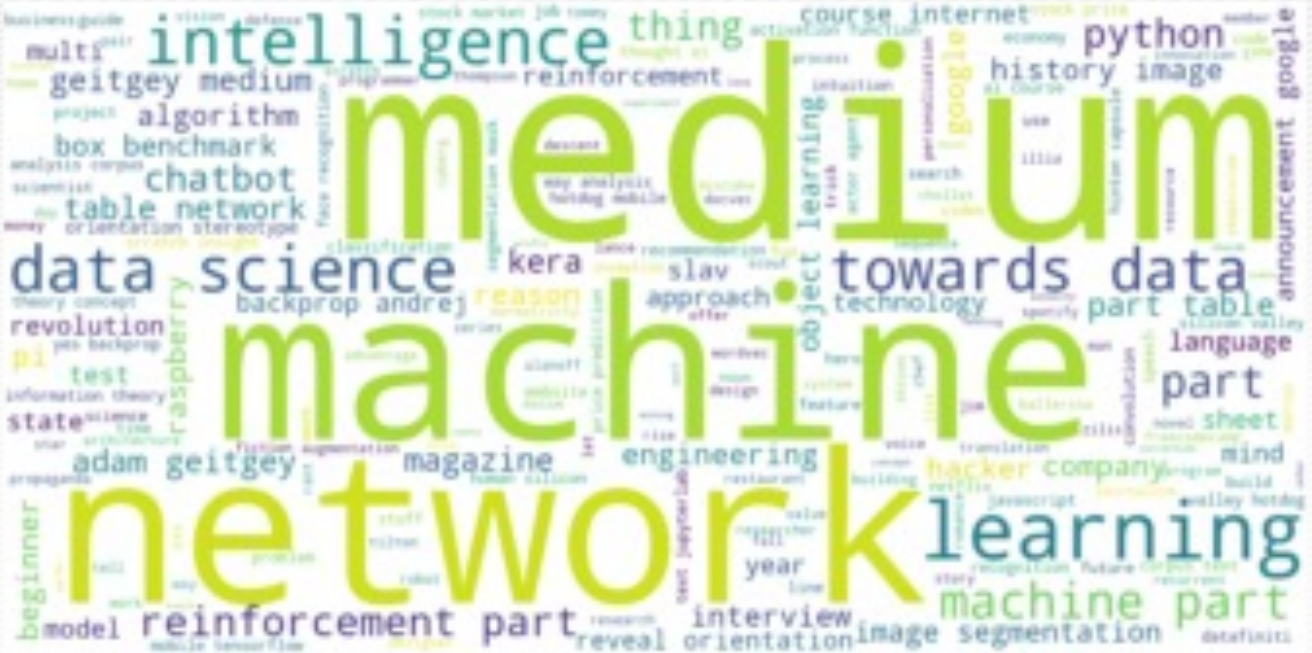
```
wordcloud = wc.WordCloud(width=1600, height=800, max_words=100,
                         background_color="white").generate(text)
plt.imshow(wordcloud)
plt.axis("off")
```

```
(-0.5, 1599.5, 799.5, -0.5)
```



After advanced processing…





03

K-MOOC 인문 • 사회계열전공생을 위한 Python 입문

성신여자대학교 SUNGSHIN UNIVERSITY

K-MOOC

성신여대가 한국형 온라인 공개강좌(K-MOOC) 운영기관 중
블루리본을 비롯한 최우수강좌에 최다 선정되었습니다.

K-MOOC(Korean Massive Open Online Course)는
교육부가 주관하고 국가평생교육원이 시행하는 온라인
공개강좌 서비스로, 매년 운영강좌를 대상으로 연차평가를
시행하여 우수한 강좌에 우수 등급(S 매우 우수, A 우수)을
부여하고 S등급 강좌 중 최우수강좌를 선정하여
블루리본을 수여하고 있습니다.

블루리본 강좌는 K-MOOC 홈페이지에 방문하는 수강생들에게
특별히 추천하는 강좌로서 선정강좌에는 블루리본을 달아
표시합니다.

성신여자대학교는 ▲이정윤 교수의 「현대인의 정신건강」
▲윤태진 교수의 「인문·사회계열 전공생을 위한 Python 입문」
▲김용재 교수의 「우리 문화 속의 한자어」
▲강경선 교수의 「음악은 왜 치료적인가」 총 4개 강좌가
블루리본 강좌로 선정되어 운영기관 중 최다 선정되는
영예를 안았습니다.

K-MOOC 인문 • 사회계열전공생을 위한 Python 입문

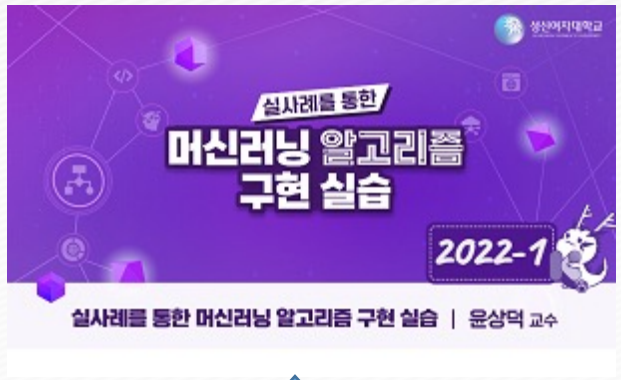
특히 윤태진 교수의 「인문·사회계열 전공생을 위한 Python 입문」 강좌는 개발 첫해에 블루리본 강좌로 선정되는 쾌거를 이루어 냈으며, 김용재 교수의 「우리 문화 속의 한자어」와 강경선 교수의 「음악은 왜 치료적인가」 두 강좌 모두 2회째 블루리본을 수여하는 대기록을 달성하였습니다.

또한 소현진 교수의 「설득의 과학」, 김연식 교수의 「헌법: 갈등해결의 코드」도 최우수강좌로 선정되었습니다. 소현진 교수의 「설득의 과학」은 블루리본과 교육부장관표창을 수상한 강좌로, 개발 이래 매년 우수성을 인정받고 있습니다.

K-MOOC 인문•사회계열전공생을 위한 머신러닝 예비학교

성신여자대학교 SUNGSHIN UNIVERSITY

K-MOOC



THANK YOU