

```
/* elice */
```

# 파이썬 실전 데이터 분석

# 텍스트 파일 분석



# 커리큘럼

1 ○

## 트럼프 대통령 트윗으로 알아보는 미국의 주요 정치 이슈

조건문, 반복문, 리스트를 복합적으로 활용해 트럼프 대통령의 트윗을 분석하고, 알아보기 쉬운 시각화로 표현해 봅니다.

2 ○

## 영어 단어 모음으로 시작하는 데이터 시각화

파이썬 라이브러리와 파일 불러오기를 이용해 영문 자료에서 가장 많이 사용되는 단어 10,000개를 알아봅니다.

# 목차

1. 파일 다루기
2. 데이터 구조 다루기
3. 그래프 다루기

# 파일 다루기

# 파일 열기/닫기

```
file = open('data.txt')  
content = file.read()  
file.close()
```

# 파일 자동으로 닫기

```
with open('data.txt') as file:  
    content = file.read()  
# file.close() - 필요 없음
```

# 줄 단위로 읽기

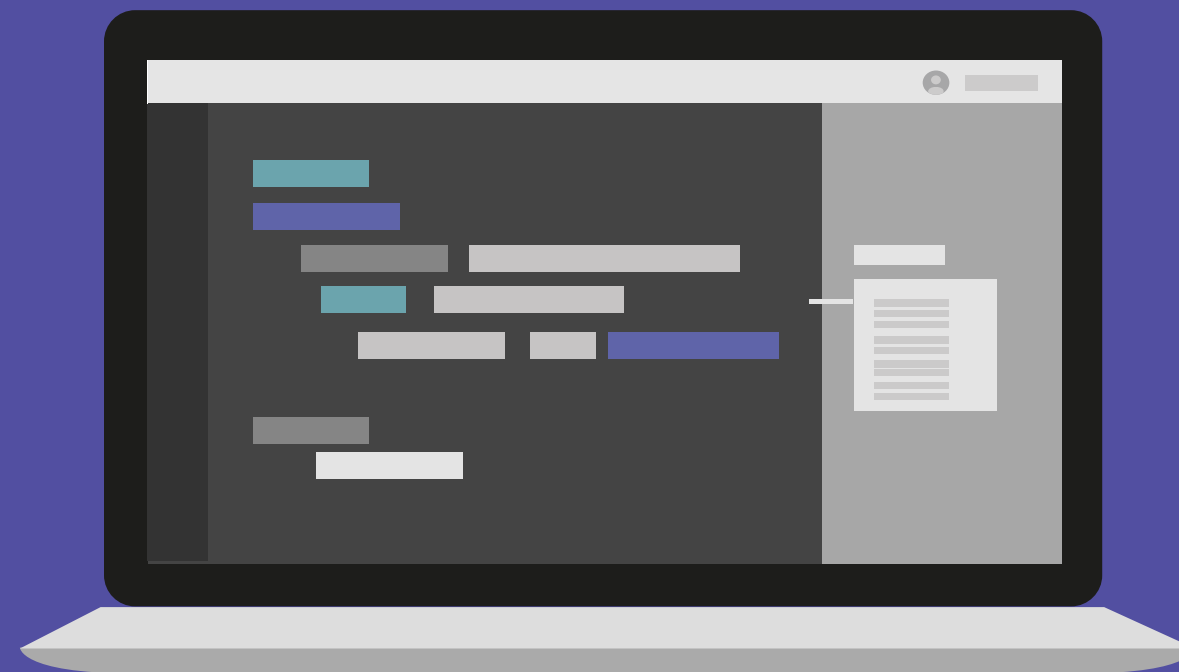
```
contents = []  
with open('data.txt') as file:  
    for line in file:  
        contents.append(line)
```

# 파일의 모드

```
# 쓰기 (Write) 모드로 파일을 연다  
with open('data.txt', 'w') as file:  
    file.write('Hello')
```



# [실습1] 파일 열고 읽기



```
/* elice */
```

# 데이터 구조 다루기

# 튜플 (Tuple)

```
hello = ('안녕하세요', 'hello', 'bonjour')
```

```
apple = ('사과', 'apple', 'pomme')
```

```
red = ('빨갱다', 'red', 'rouge')
```

# 튜플 vs. 리스트

## 공통점

- 순서가 있는 원소들의 집합

## 차이점

- 각 원소의 값을 수정할 수 없음
- 원소의 개수를 바꿀 수 없음

# 리스트 변형

```
hello = ['안녕하세요', 'hello', 'bonjour']
```

```
hello[0] = '안녕'
```

```
# ['안녕', 'hello', 'bonjour']
```

```
hello.append('ni hao')
```

```
# ['안녕', 'hello', 'bonjour', 'ni hao']
```

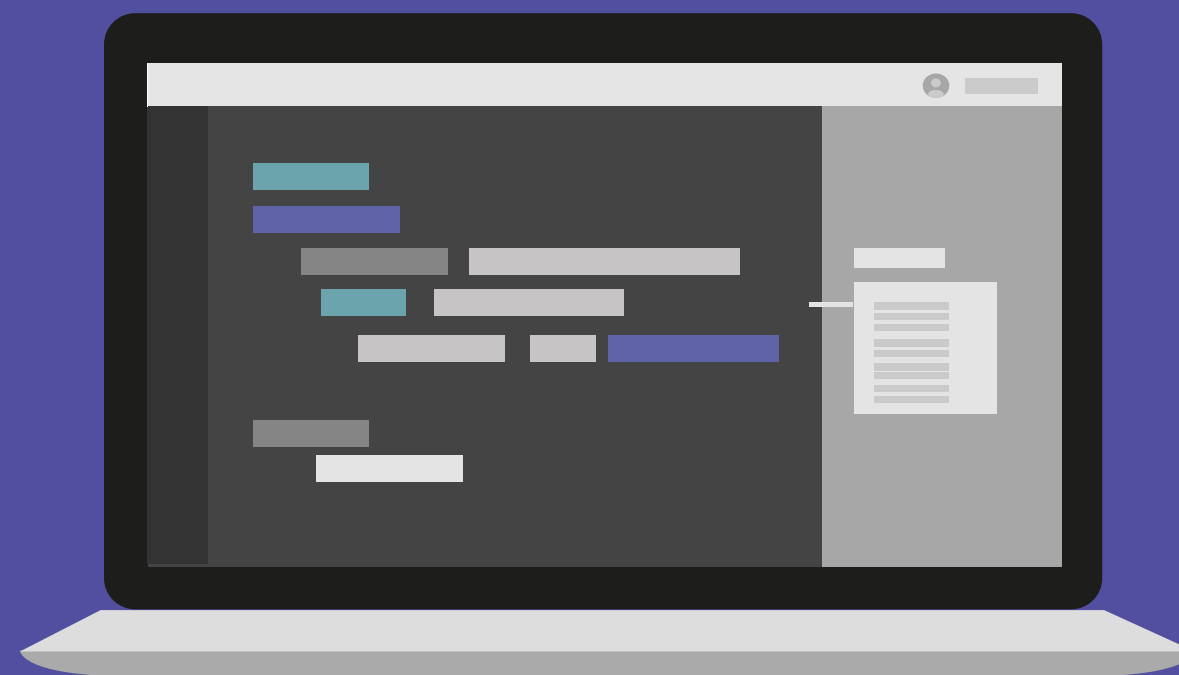
# 튜플 변형

```
hello = ('안녕하세요', 'hello', 'bonjour')
```

```
hello[0] = '안녕' # 에러
```

```
hello.append('ni hao') # 그런 함수 없음
```

# [실습2] 튜플



```
/* elice */
```

# 리스트로 리스트 만들기

```
words = ['life', 'love', 'faith']  
first_letters = []  
for word in words:  
    first_letters.append(word[0])
```



# 리스트로 리스트 만들기

```
words = ['life', 'love', 'faith']  
first_letters = [word[0] for word in words]
```

# 리스트로 리스트 만들기

```
numbers = [1, 3, 5, 7]
new_numbers = []
for n in numbers:
    new_numbers.append(n + 1)
```

# 리스트로 리스트 만들기

```
numbers = [1, 3, 5, 7]
```

```
new_numbers = [n + 1 for n in numbers]
```

# 특정 원소 걸러내기

```
numbers = [1, 3, 4, 5, 6, 7]
even = []
for n in numbers:
    if n % 2 == 0:
        even.append(n)
```

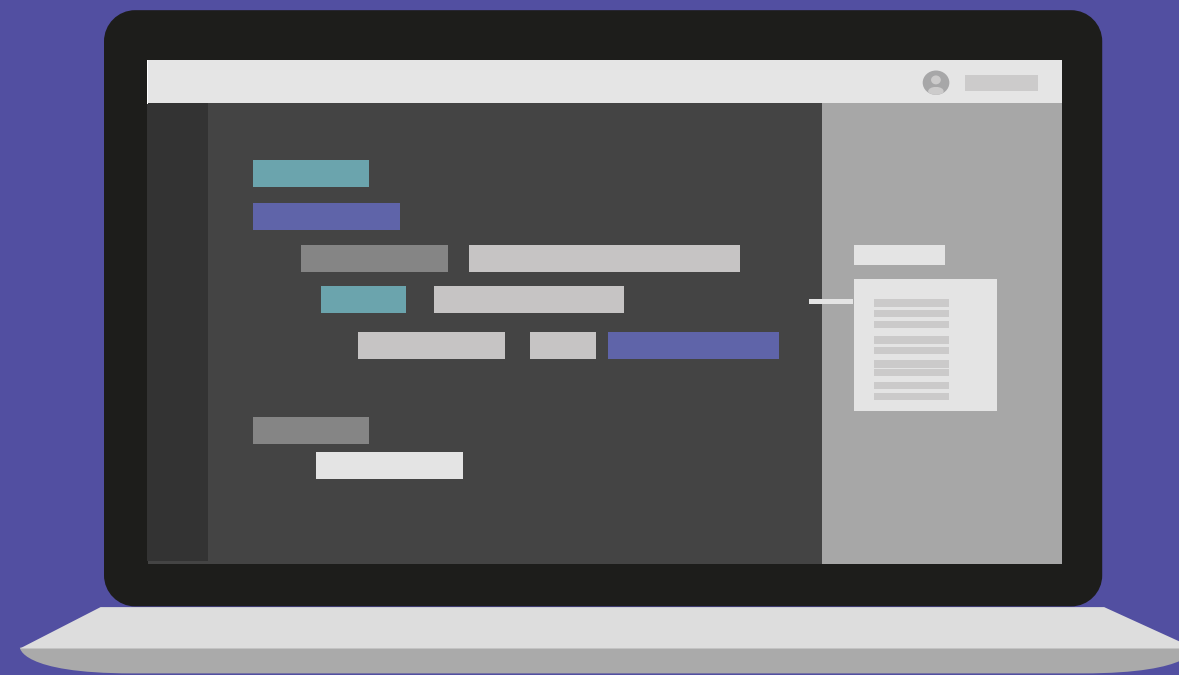
# 특정 원소 걸러내기

```
numbers = [1, 3, 4, 5, 6, 7]  
even = [n for n in numbers if n % 2 == 0]
```

# 특정 원소 걸러내기

```
numbers = [1, 3, 4, 5, 6, 7]  
odd = [n + 1 for n in numbers if n % 2 == 0]
```

# [실습3] 한 줄로 코드 짧게 쓰기



```
/* elice */
```

# 데이터 정렬하기

```
numbers = [-1, 3, -4, 5, 6, 100]  
sort_by_abs = sorted(numbers, key=abs)
```



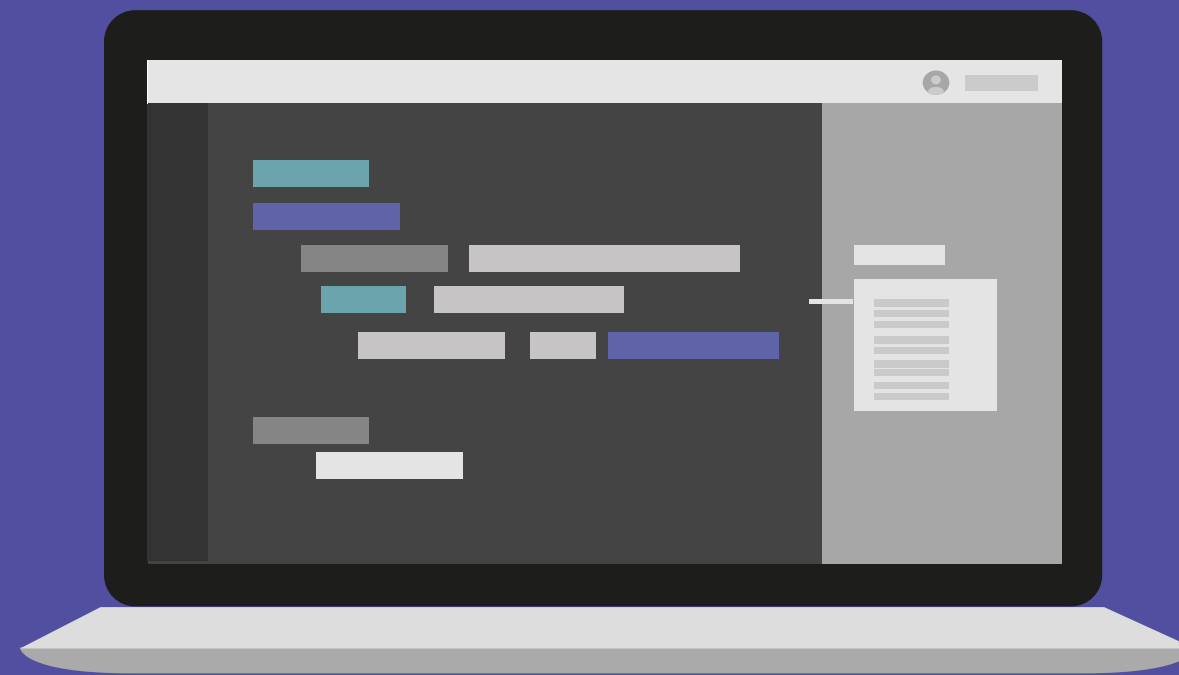
# 데이터 정렬하기

```
fruits = ['cherry', 'apple', 'banana']  
sort_by_alphabet = sorted(fruits)  
# ['apple', 'banana', 'cherry']
```

# 데이터 정렬하기

```
def reverse(word):  
    return str(reversed(word))  
  
fruits = ['cherry', 'apple', 'banana']  
sort_by_last = sorted(fruits, key=reverse)  
# ['banana', 'apple', 'cherry']
```

# [실습4] 데이터 정렬하기



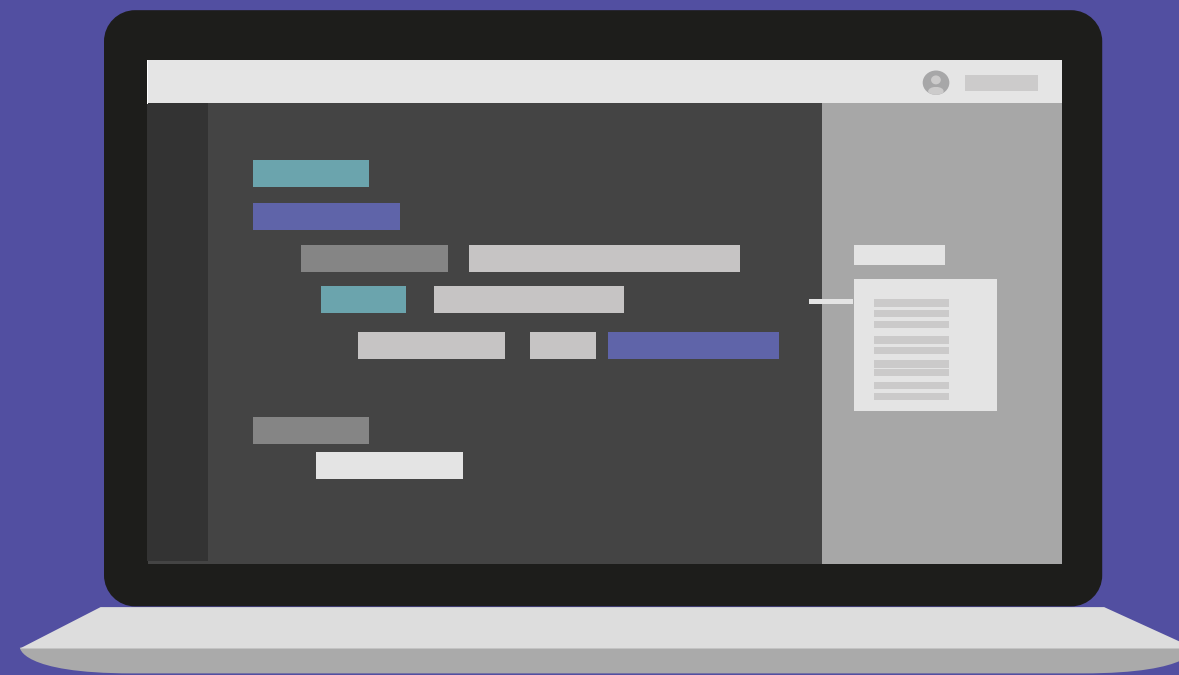
```
/* elice */
```

# 그래프 다루기

# matplotlib

- Mathematical Plot Library
- 파이썬에서 그래프를 그릴 수 있게 하는 라이브러리
- 꺾은선 그래프, 막대 그래프 등을 모두 지원

# [실습5] 차트 그리기



`/* elice */`

`/* elice */`

**문의 및 연락처**

[academy.elice.io](https://academy.elice.io)

[contact@elice.io](mailto:contact@elice.io)

[facebook.com/elice.io](https://facebook.com/elice.io)

[medium.com/elice](https://medium.com/elice)