

/\* elice \*/

# 데이터 분석 기초

김건우 선생님 · 일 14:00



# 데이터 분석 기초

목표 1 | 파이썬 문법과 프로그래밍의 개념 이해하기

목표 2 | 실제 데이터를 다루고 분석해 보기

목표 3 | 상위 수업으로 넘어갈 수 있는 기초 기르기

# 수강 대상



If문을 한 번이라도 써 보신 분



For문을 한 번이라도 써 보신 분



리스트를 한 번이라도 써 보신 분

`/* elice */`

# 주차별 커리큘럼

## 1주차

조건문, 반복문, 리스트를 복합적으로 이용해 트럼프 대통령의 트윗을 분석하고 2017 뉴욕 타임즈에서 가장 많이 읽힌 미국 정치 뉴스와 비교해봅니다.

## 2주차

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

## 3주차

넷플릭스 데이터 대회에 사용된 JSON 데이터 일부를 사전형으로 변환하고 재정렬해보며 비슷한 성향의 유저를 찾아봅니다.

## 4주차

CSV, JSON, TXT 등 다양한 형태의 데이터와 앞서 배운 개념을 종합적으로 다루며 테드 강연에서 가장 인기 있는 동영상과 핫한 주제를 분석·시각화 해봅니다.

/\* elice \*/

# 수업의 구성



녹화영상  
실습문제

개념/문법 습득



라이브 강의

프로젝트 진행

# 선생님 소개



## 김건우 선생님

- KAIST 전산학부
- (전) 엘리스 <코딩학교> 강사
- (전) SK Hynix 소프트웨어 개발 인턴
- (전) KAGE 영재학술교육원 강사

# 1주차: 트럼프 대통령 트윗 분석

1. 리스트 순회하기
2. 파이썬의 문자열 관련 함수

# 01 리스트 순회하기

`/* elice */`



# for 반복문

```
fruits = [“사과”, “바나나”, “키위”]
```

```
# 리스트의 원소를 하나씩 가져옵니다.
```

```
for fruit in fruits:
```

```
    print(fruit + “는 맛있어”)
```

# for 반복문

# 0부터 9까지 한 줄 씩 출력합니다.

```
for num in range(10):  
    print(num)
```

# for 반복문

```
fruits = ["사과", "바나나", "키위"]
```

```
# 과일의 이름과 번호를 함께 출력합니다.
```

```
for i in range(len(fruits)):
    print("과일" + str(i+1) + ":" + fruits[i])
```

# 실습 1: for, in



```
/* elice */
```

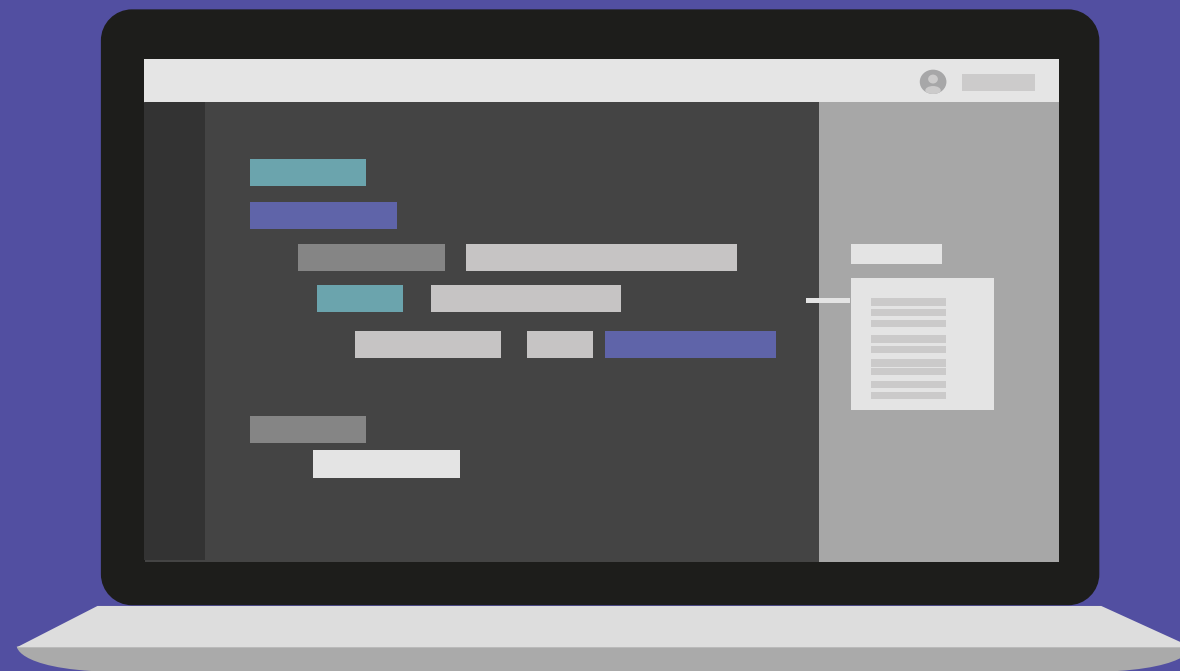
# 인덱싱

```
fruits = ["사과", "바나나", "키위", "배"]  
last_fruit = fruits[-1]  
tropical_fruits = fruits[1:3]  
no_apple = fruits[1:]  
no_pear = fruits[:3]
```

# 문자열 인덱싱

```
word = "superman"
print(word[3])           # 'e'
print(word[-2])          # 'a'
print(word[5:])          # 'man'
print(word[:5])          # 'super'
```

# 실습 2: 인덱싱



`/* elice */`

# 02 문자열 함수



# .startswith()

```
word = "superman"  
print(word.startswith('s'))      # True  
  
if word.startswith('a'):  
    print("a로 시작하는 단어입니다.")
```

# 실습 3: `startswith()`



```
/* elice */
```

# .split()

```
intro = “제 이름은 엘리스입니다.”
```

```
print(intro.split())
```

```
>>> [“제”, “이름은”, “엘리스입니다.”]
```

```
fruits = “사과,귤,배,바나나”
```

```
print(fruits.split(','))
```

```
>>> [“사과”, “귤”, “배”, “바나나”]
```

# .split()

```
numbers = " 1 2 3 "
```

```
print(numbers.split())
```

```
>>> ['1', '2', '3']
```

```
print(numbers.split(' '))
```

```
>>> ['', '', '1', '', '2', '', '3', '', '']
```

# 대표적인 공백 문자

‘ ’

빈칸  
(스페이스바)

‘\t’

Tab  
(Tab 키)

’\n’

Newline  
(엔터 키)

# 실습 4: split()



```
/* elice */
```

# .append()

```
numbers = []  
numbers.append(1)  
print(numbers)           # [1]  
numbers.append(2)  
print(numbers)           # [1, 2]
```

# .append()

```
numbers = [1, 2, 10, 17]
```

```
small_numbers = []
```

```
for num in numbers:
```

```
    if number < 10:
```

```
        small_numbers.append(num)
```



# 실습 5: append()



```
/* elice */
```

# 대소문자 변환

```
intro = "My name is Elice!"
```

```
print(intro.upper())
```

```
>>> "MY NAME IS ELICE!"
```

```
print(intro.lower())
```

```
>>> "my name is elice!"
```

# append()와 lower()의 차이

```
words = ['hello']  
words.append('elice')  
print(words)  
>>> ['hello', 'elice']
```

/\* elice \*/

# append()와 lower()의 차이

```
intro = "My name is Elice"  
intro.lower()  
print(intro)  
>>> ???
```

# append()와 lower()의 차이

```
intro = "My name is Elice"
```

```
intro.lower()
```

원래 문자열을 직접 수정하지 않는다!

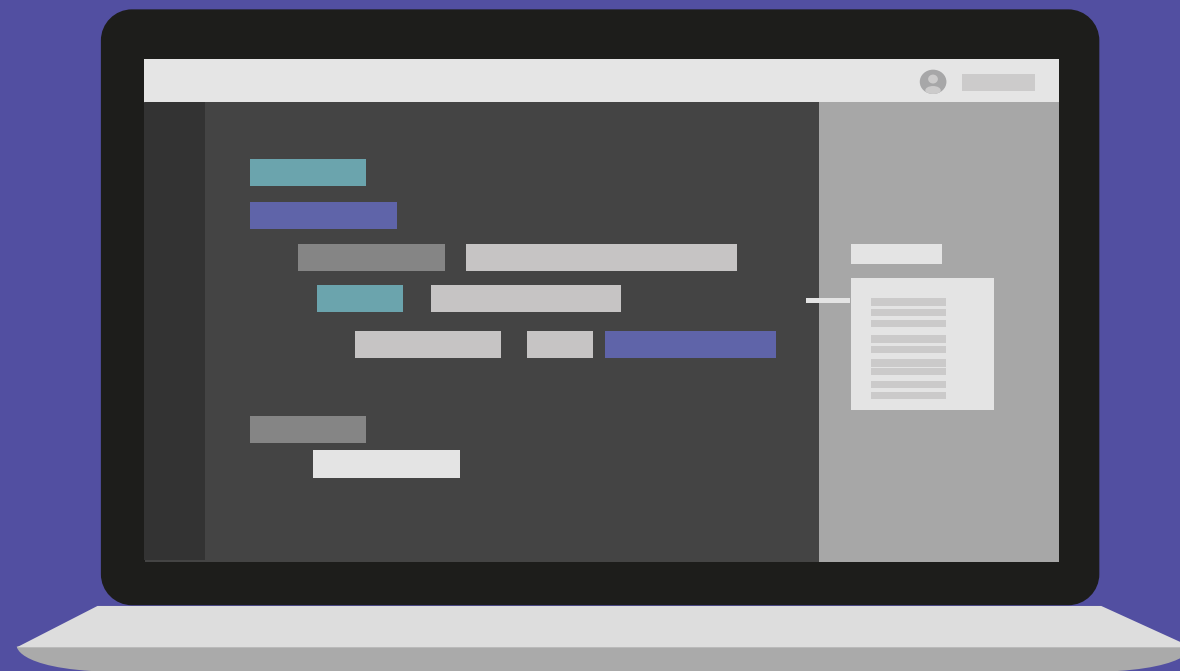
```
print(intro)
```

```
>>> "My name is Elice"
```

# append()와 lower()의 차이

```
intro = "My name is Elice"  
lower_intro = intro.lower()  
print(lower_intro)  
>>> "my name is elice"
```

# 실습 6: lower()



```
/* elice */
```

# .replace()

```
intro = “제 이름은 Elice입니다.”  
print(intro.replace(‘Elice’, ‘엘리스’))  
>>> “제 이름은 엘리스입니다.”
```



# .replace()

```
intro = “제 이름은 Elice입니다.”  
print(intro.replace(' ', ''))  
>>> “제이름은Elice입니다.”
```

# .replace()

```
intro = “제 이름은 Elice입니다.”
```

```
intro.replace(' ', ‘ ’)
```

```
print(intro)
```

```
>>> “제 이름은 Elice입니다.”
```

# 실습 7: replace()



```
/* elice */
```

`/* elice */`

문의 및 연락처

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

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

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

[blog.naver.com/elicer](http://blog.naver.com/elicer)