

```
/* elice */
```

# 파이썬 실전 데이터 분석

# 데이터 처리

# 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.

# 파일 다루기

# 파일 열기/닫기

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

# with

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

# 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]")
```

# [실습1] for, in



```
/* elice */
```



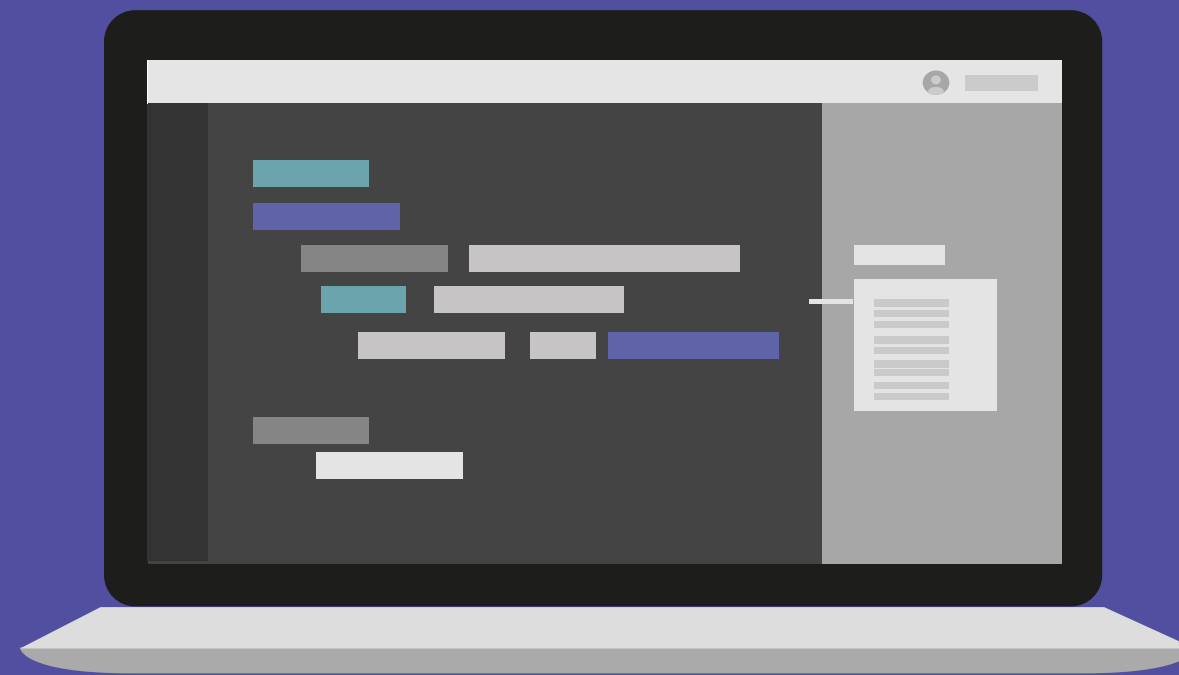
# 인덱싱

```
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 */
```

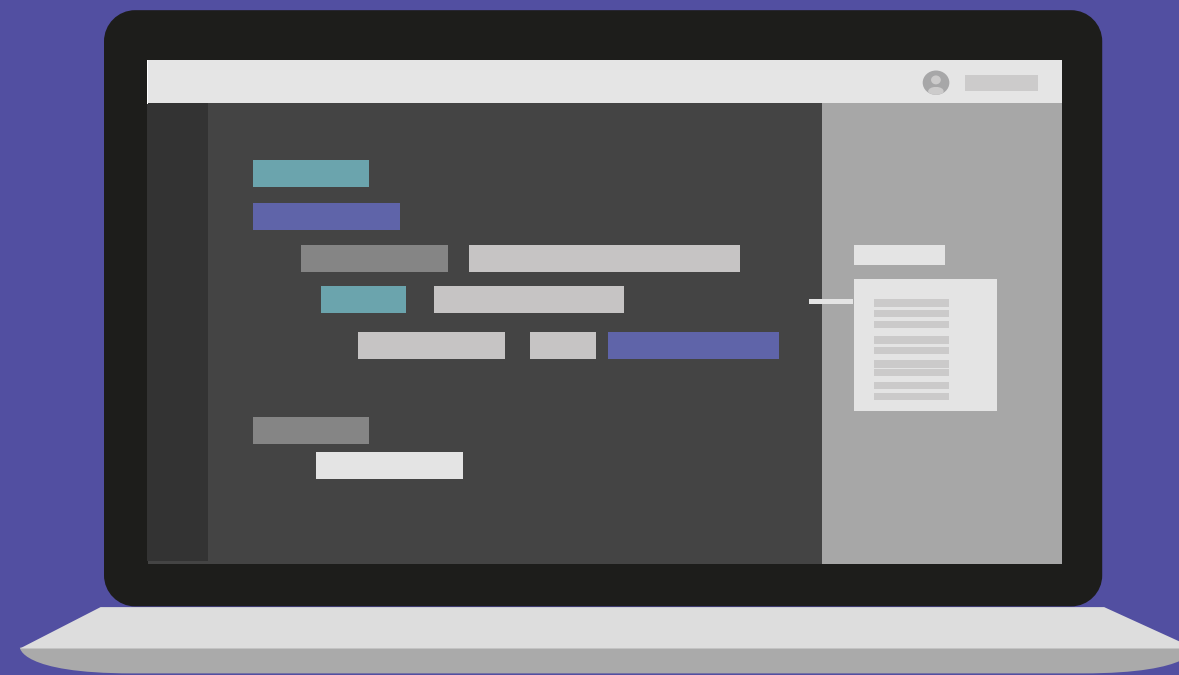
# 문자열 함수

# .startswith()

```
word = "superman"
print(word.startswith('s'))      # True

if word.startswith('a'):
    print("a로 시작하는 단어입니다.")
```

# [실습3] 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)

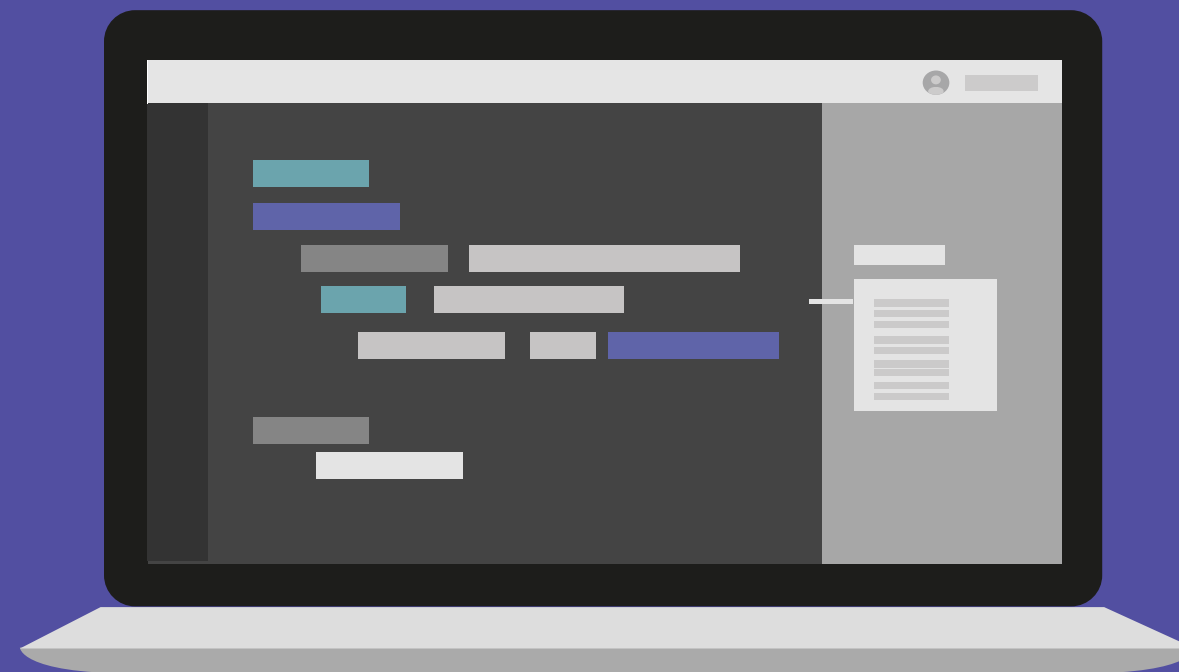
'\t'

Tab  
(Tab 키)

'\n'

Newline  
(엔터 키)

# [실습4] 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 num < 10:
        small_numbers.append(num)
```

# [실습5] 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']
```

# 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!

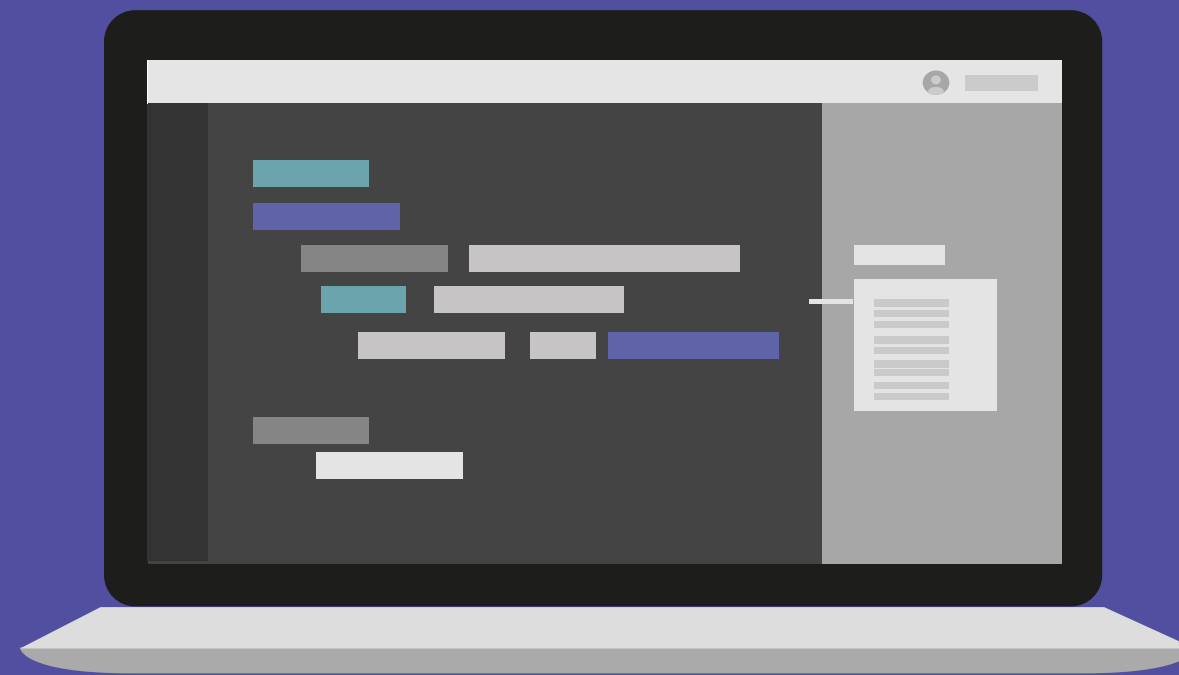
```
print(intro)
```

```
>>> "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"
```

# [실습6] lower()



```
/* elice */
```

# .replace()

```
intro = "제 이름은 Elice입니다."  
print(intro.replace('Elice', '엘리스'))  
>>> "제 이름은 엘리스입니다."
```

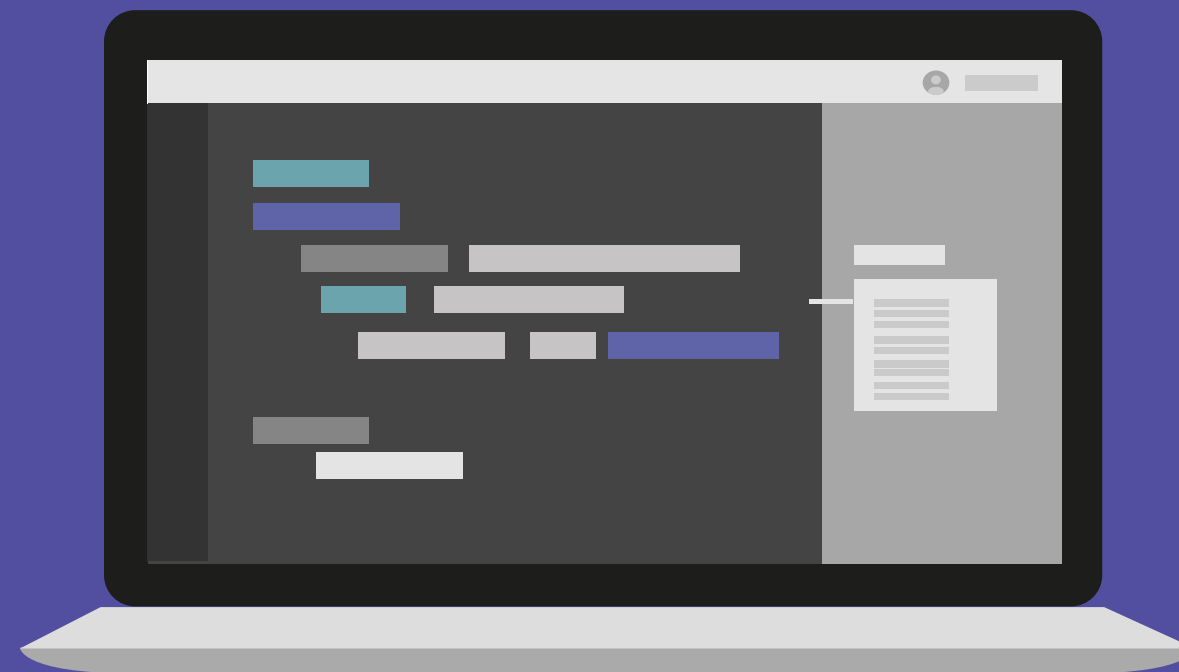
# .replace()

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

# .replace()

```
intro = "제 이름은 Elice입니다."  
intro.replace(' ', '')  
print(intro)  
>>> "제 이름은 Elice입니다."
```

# [실습7] replace()



```
/* 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)