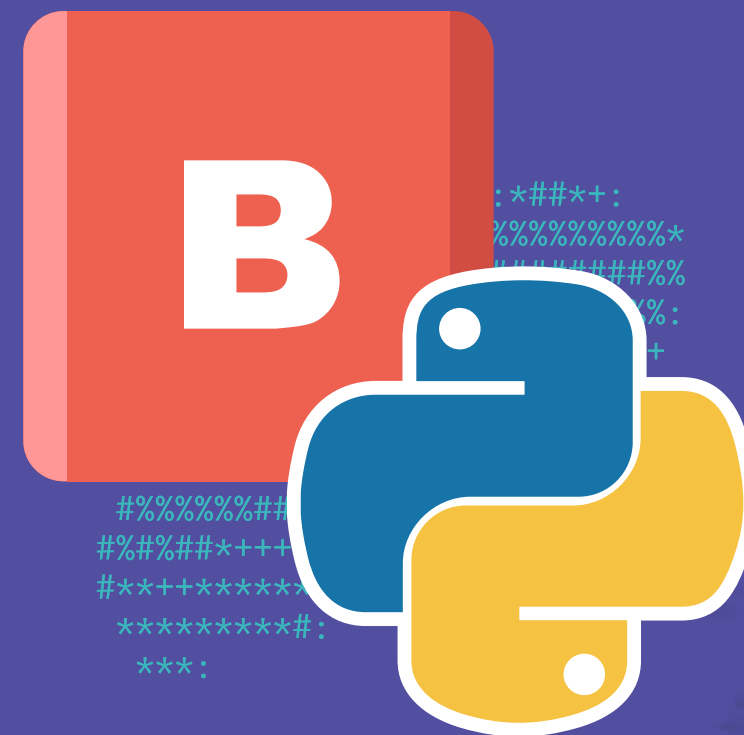


/\* elice \*/

# 도레미 파이썬 Vol.2

기초 자료형 II



이호준 선생님

# 수강 목표

문자열과 리스트를 더 활용할 수 있다

튜플이 무엇인지 알고, 리스트와의 차이점을 이해한다

딕셔너리를 이해하고 이를 직접 만들 수 있다

# 목차

1. 문자열/리스트 활용
2. Tuple(튜플)
3. Dictionary(딕셔너리)

# 문자열 / 리스트 활용

# list.pop(i)

인덱스 **i**의 원소를 **제거** 후 그 원소를 **반환**  
(괄호를 비울 시 **마지막 원소**)

```
my_list = [1, 2, 3, 4, 5]  
print(my_list.pop(0)) # 1  
print(my_list.pop()) # 5
```

# seq.count(d)

시퀀스 내부의 자료 d의 개수를 반환

```
my_seq = [2, 2, 2, 4, 4]  
print(my_seq.count(2)) # 3
```

# str.split(c)

**c**를 기준으로 문자열을 쪼개서 리스트를 반환  
(괄호를 비울 시 **공백**)

```
my_str = "1 2 3 4 5"
```

```
print(my_str.split()) # ['1', '2', '3', '4', '5']
```

```
element = "Na,Mg,Al,Si"
```

```
print(element.split(',')) # ['Na', 'Mg', 'Al', 'Si']
```

# str.join(list)

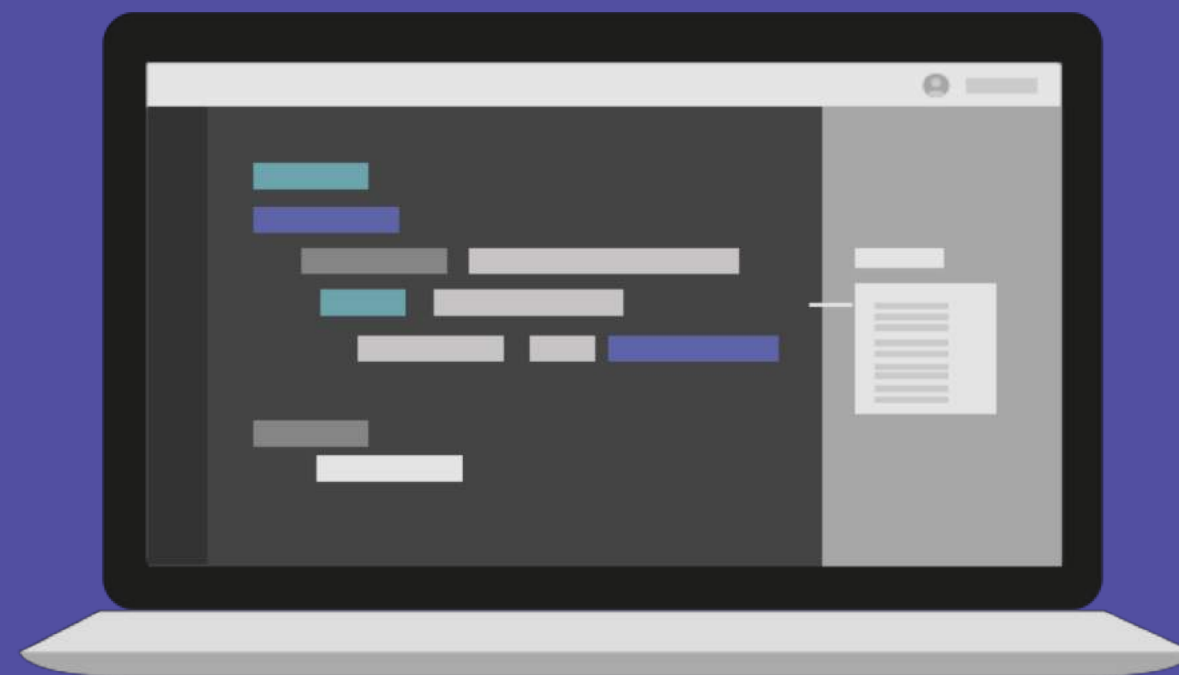
**str**을 **기준**으로 리스트를 합쳐서 문자열을 반환  
(괄호를 비울 시 **공백**)

```
my_list = ['a', 'p', 'p', 'l', 'e']  
print(''.join(my_list)) # apple  
friend = ['Pat', 'Mat']  
print('&'.join(friend)) # Pat&Mat
```



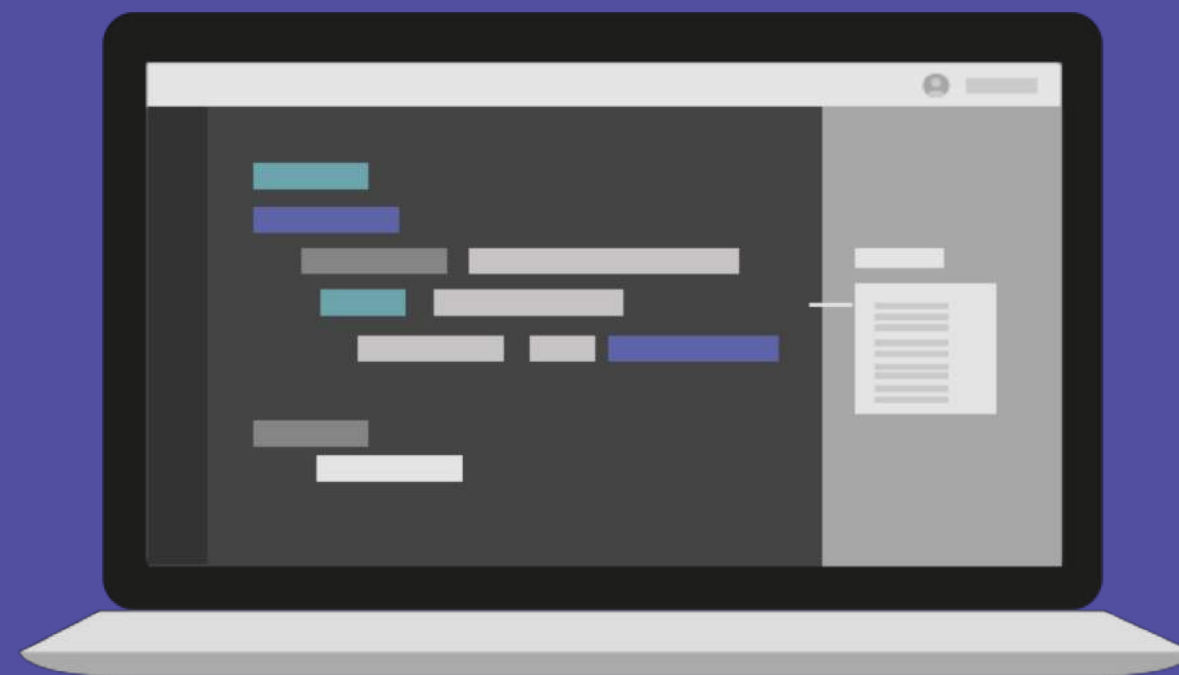
[실습1]

pop, count



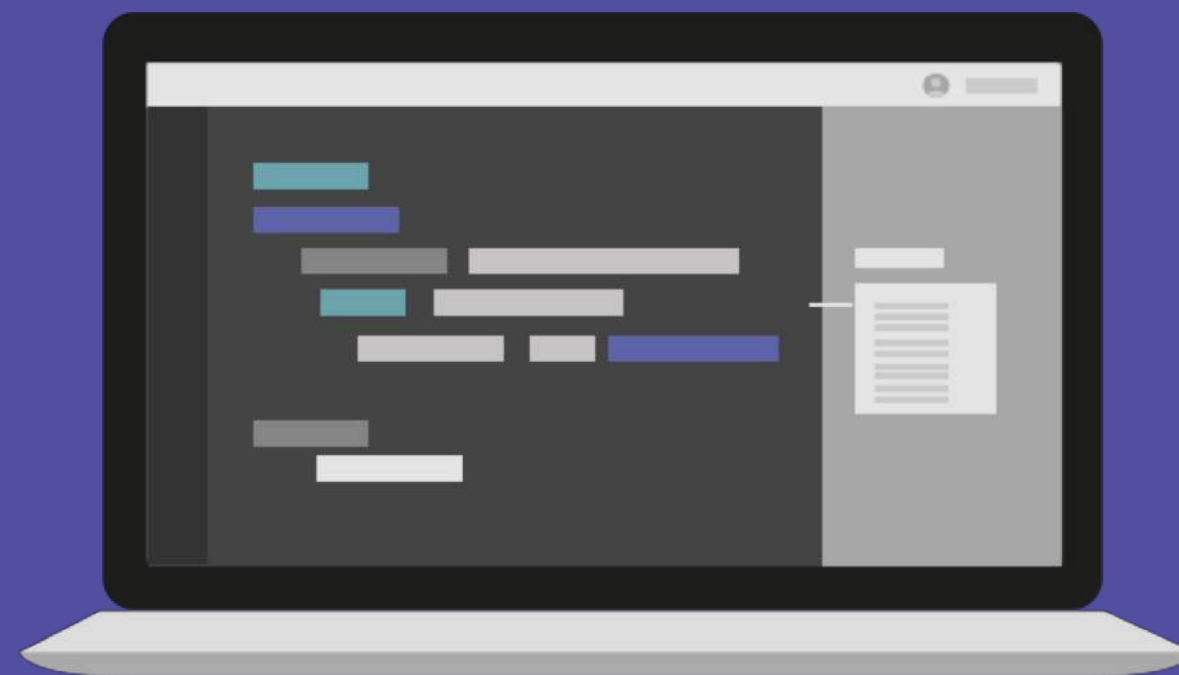
# [실습2]

## split, join



# [실습3]

## 노래 가사 분석



# Tuple(튜플)

# 여러 자료를 담는 자료형이 필요하면?

대부분 **리스트**를 이용  
그러나 **값이 바뀔 위험**이 있다!

```
my_list = ['l', 'i', 's', 't']  
my_list[1] = 'a'  
print(my_list) # ['l', 'a', 's', 't']
```

# Tuple의 필요성

값을 바꿀 수 없으면서도  
여러 자료를 담을 순 있을까?

→ Tuple(튜플)!

# Tuple(튜플)

여러 자료를 함께 담을 수 있는 자료형

() - 소괄호로 묶어서 표현

```
tuple_zero = ()
```

```
tuple_one = (1,)
```

```
tuple_ = (1, 2, 3, 4, 5)
```

```
tuple_ = 1, 2, 3, 4, 5
```

# Tuple의 특징

시퀀스 자료형으로

Index를 이용한 **인덱싱**, **슬라이싱**이 가능

```
my_tuple = ('t', 'w', 'i', 'c', 'e')  
print(my_tuple[1]) # 'w'  
print(my_tuple[2:4]) # ('i', 'c')
```



# Tuple의 특징

**in** 연산자로 Tuple 안에 원소 확인

**len()** 함수로 Tuple의 길이 확인

```
my_tuple = ('t', 'w', 'i', 'c', 'e')  
print('t' in my_tuple) # True  
print(len(my_tuple)) # 5
```

# Tuple의 특징

- + 연산자로 Tuple과 Tuple을 연결
- \* 연산자로 Tuple을 반복

```
my_tuple = ('i', 'c', 'e')
```

```
print(('e', 'l') + my_tuple) # ('e', 'l', 'i', 'c', 'e')
```

```
print(my_tuple * 2) # ('i', 'c', 'e', 'i', 'c', 'e')
```

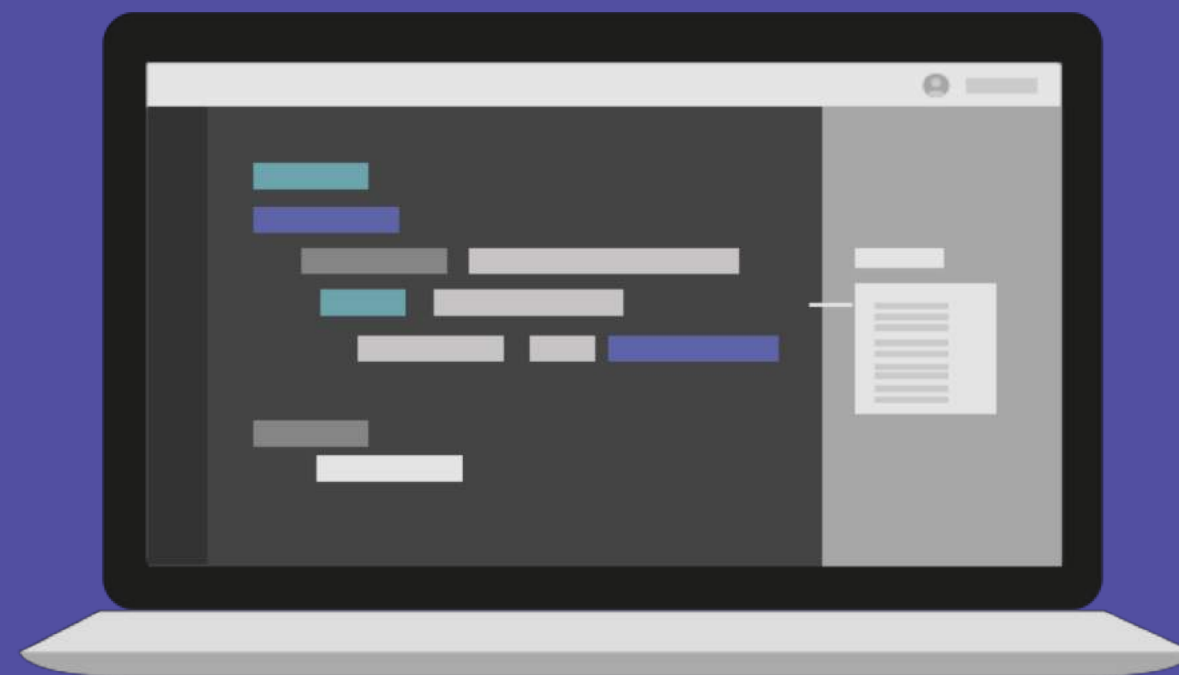
# Tuple의 특징

자료 추가, 삭제, 변경 불가  
한 번 만들어지면 고정!

```
my_tuple = ('t', 'w', 'i', 'c', 'e')  
print(my_tuple.append('!')) # Error  
print(my_tuple.remove('w')) # Error  
my_tuple[1] = 's' # Error
```

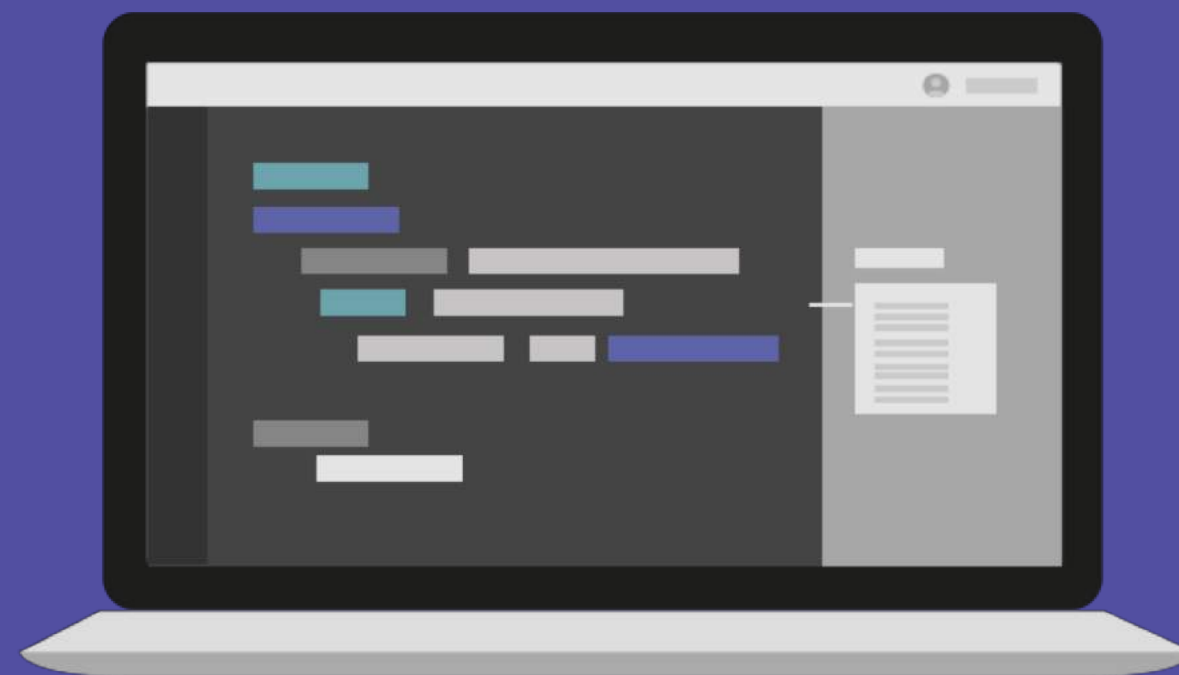
# [실습4]

## Tuple 만들기



[실습5]

# Tuple vs List



# Dictionary(딕셔너리)

# Dictionary?

Dictionary → 사전  
짜깁이 있는 자료형!

## dictionary

*noun* • **UK**  /'dɪkʃən·ər-i/ **US**  /'dɪkʃə·ner-i/ **PLURAL dictionaries**

★ **A1** a book that contains a list of words in alphabetical order with their meanings explained and sometimes written in another language

사전

*Use your dictionaries to look up any words you don't understand.*

성	이름
이메일	
비밀번호	
비밀번호 확인	

# Dictionary(딕셔너리)

{ } - 중괄호로 묶어서 표현

```
dict_zero = {}
```

```
person = {'name': 'Michael', 'age': 10}
```



# Dictionary(딕셔너리)

짝꿍이 있는 자료형

{key : value}의 형식 : **key**를 알면 **value**를 알 수 있음

```
dict_zero = {}
```

```
person = {'name': 'Michael', 'age': 10}
```

# Key

열쇠처럼 자료를 꺼낼 수 있는 도구

```
dict_zero = {}
```

```
person = {'name': 'Michael', 'age': 10}
```

# Value

Dictionary에서 Key로 꺼낸 자료

```
dict_zero = {}
```

```
person = {'name': 'Michael', 'age': 10}
```

# Dictionary[key]

Dictionary에서 자료를 꺼내기

```
person = {'name': 'Michael', 'age': 10}
print(person['name']) # Michael
print(person['age']) # 10
```

Dictionary   key

# Dictionary[key]

Dictionary에서 자료를 **추가하기**

```
person = {'name': 'Michael', 'age': 10}
```

```
person['hometown'] = Seoul
```

Dictionary

key

value

# del

## del 함수로 Dictionary의 원소 삭제

```
person = {'name': 'Michael', 'age': 10}
```

```
del person['age']
```

Dictionary   key

```
print(person) # {'name': 'Michael'}
```

# Dictionary의 특징

Key는 변할 수 없는 자료형

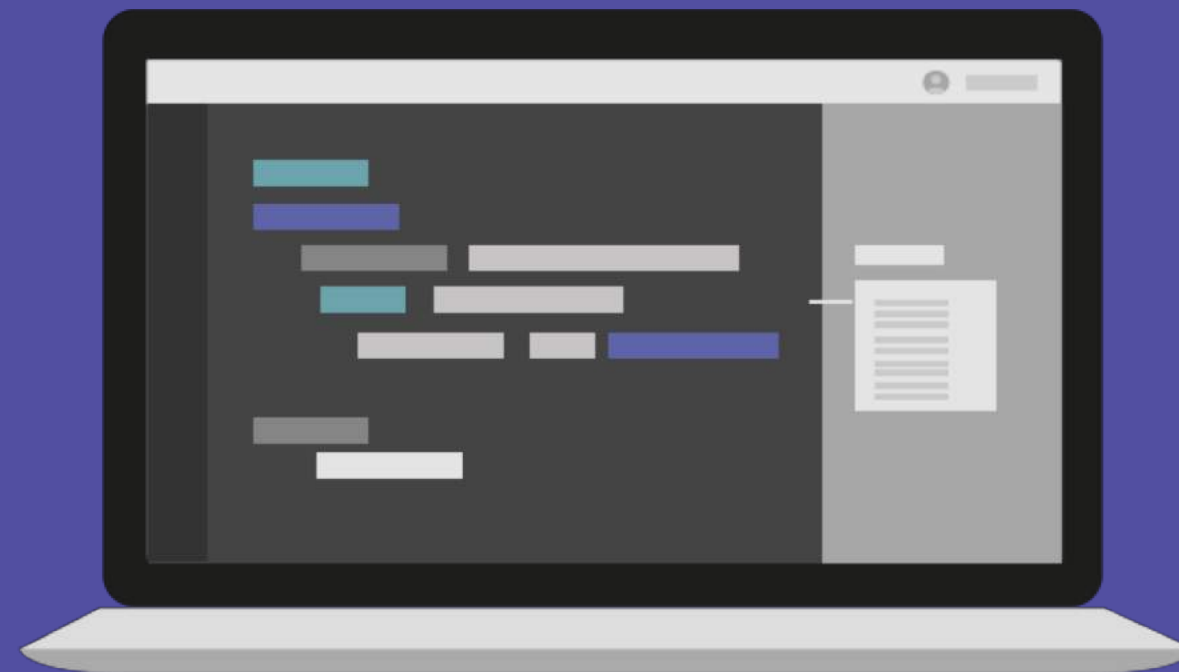
→ 리스트는 안되고, 튜플은 된다!

```
datas = {[1, 2, 3]: 'Alphabet'} # Error
```

```
datas = {(1, 2, 3): 'Number'} # OK
```

[실습6]

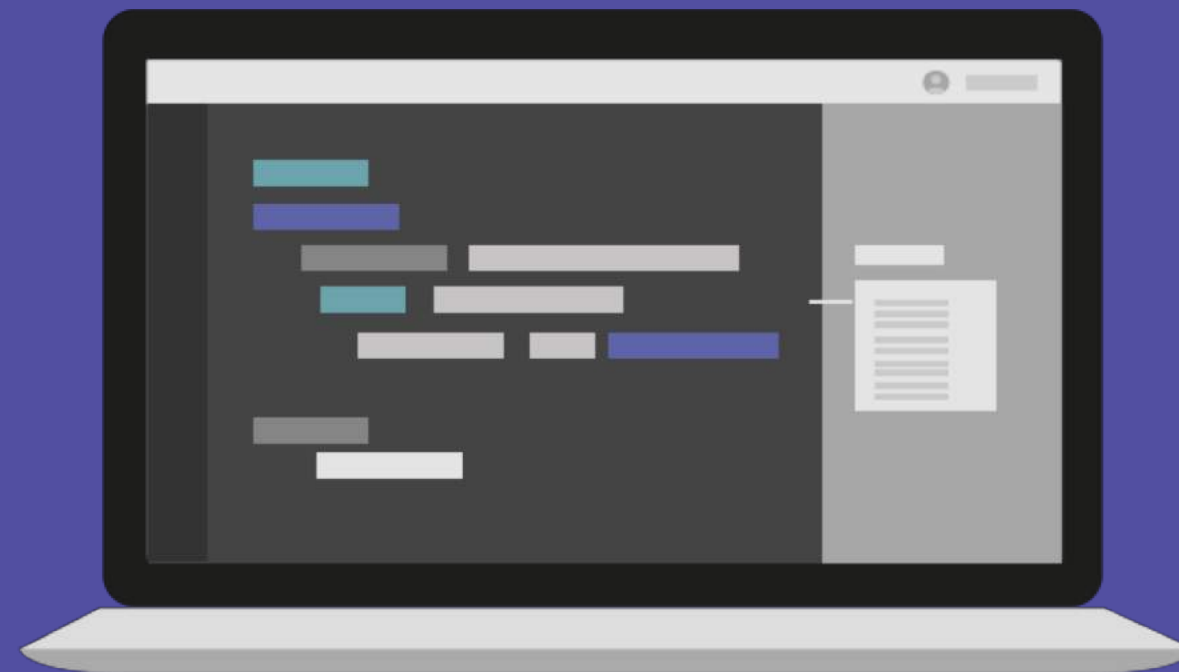
# Dictionary 만들기





[실습7]

Dictionary의 key



# 요약

# 요약

문자열과 리스트의 활용

**Tuple**과 리스트의 차이점을 통해 그 특성을 이해

**Dictionary** 속 Key와 Value 짝꿍

```
list.pop()
```

```
str.split()
```

```
sequence.count()
```

```
str.join()
```

`/* 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)