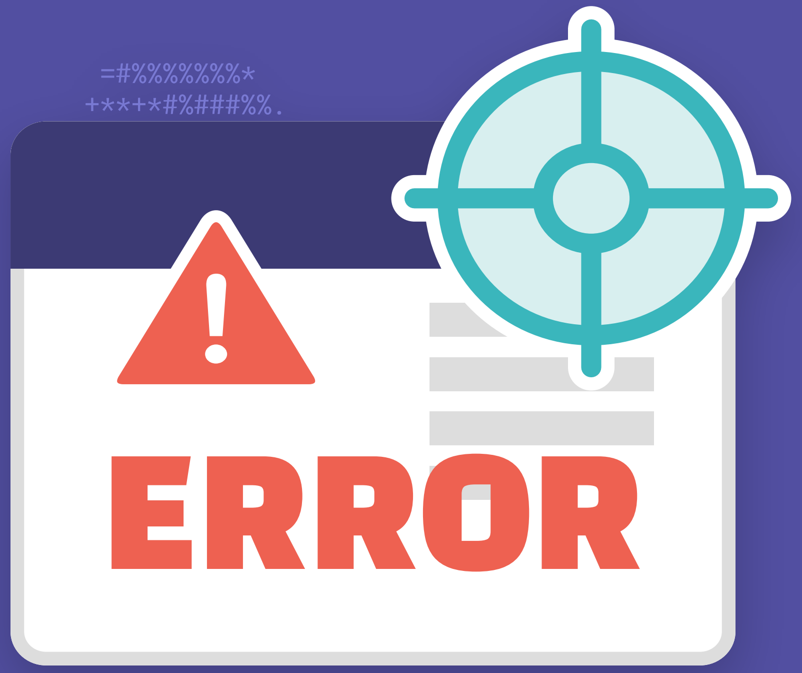
/* elice */

도전! 디버깅 입문

에러를 만나도 당황하지 않는 법



김건우 선생님

커리큘럼

1

왜 디버깅이 필요할까요?

프로그래머와 사용자가 겪을 수 있는 다양한 상황을 이해합니다.

디버깅이 무엇인지, 디버깅이 왜 중요한지 배웁니다.

2

에러 읽고 대처하기

코딩을 하면서 만나는 다양한 에러 코드를 이해하고,

이에 대처하는 방법을 배워 봅니다.

에러를 마주했을 때의 공포감을 극복합니다.

목차

1. 에러 메시지 읽기
2. 자주 접하는 에러 코드
3. try / except

에러 메시지 읽기

에러 메시지 읽기

```
def greeting(your_name):
    print("Hello, " + yourname + "!”)

greeting("Donald Trump”)
```

에러 메시지 읽기

```
Traceback (most recent call last):
  File “main.py”, line 4, in <module>
    greeting(“Donald Trump”)
  File “main.py”, line 2, in greeting
    print(“Hello”, yourname + “!”)
NameError: name ‘yourname’ is not defined
```

에러 메시지 읽기

```
Traceback (most recent call last):
  File "main.py", line 4, in <module>
    greeting("Donald Trump")
  File "main.py", line 2, in greeting
    print("Hello", yourname + "!")
NameError: name 'yourname' is not defined
```

에러 메시지 읽기

```
Traceback (most recent call last):
  File ”main.py”, line 4, in <module>
    greeting(”Donald Trump”)
  File ”main.py”, line 2, in greeting
    print(”Hello”, yourname + ”!”)
NameError: name ’yourname’ is not defined
```

에러 메시지 읽기

```
Traceback (most recent call last):
  File "main.py", line 4, in <module>
    greeting("Donald Trump")
  File "main.py", line 2, in greeting
    print("Hello", yourname + "!")
NameError: name 'yourname' is not defined
```

에러 메시지 읽기

```
Traceback (most recent call last):
  File ”main.py”, line 4, in <module>
    greeting(”Donald Trump”)
  File ”main.py”, line 2, in greeting
    print(”Hello”, yourname + ”!”)
NameError: name ’yourname’ is not defined
```

에러 메시지 읽기

```python
def average(numbers):
    return sum(numbers) / len(numbers)

average([])
```

에러 메시지 읽기

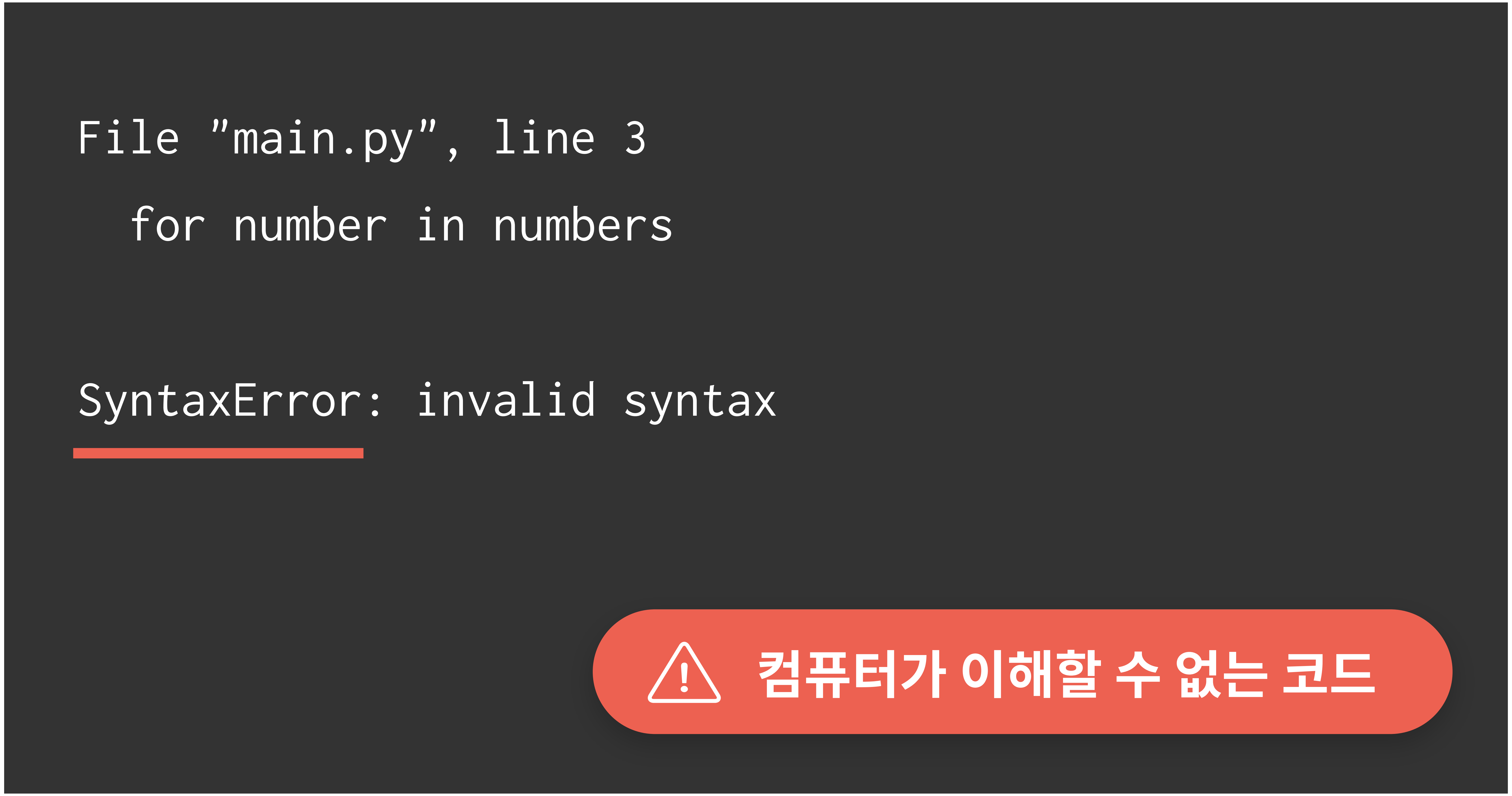
```
Traceback (most recent call last):
  File ”main.py”, line 4, in <module>
    average([])
  File ”main.py”, line 2, in average
    return sum(numbers) / len(numbers)
ZeroDivisionError: division by zero
```

자주 접하는 에러 코드

Syntax error

```
def add_all(numbers):
    result = 0
    for number in numbers
        result += number
    return result
```

Syntax error



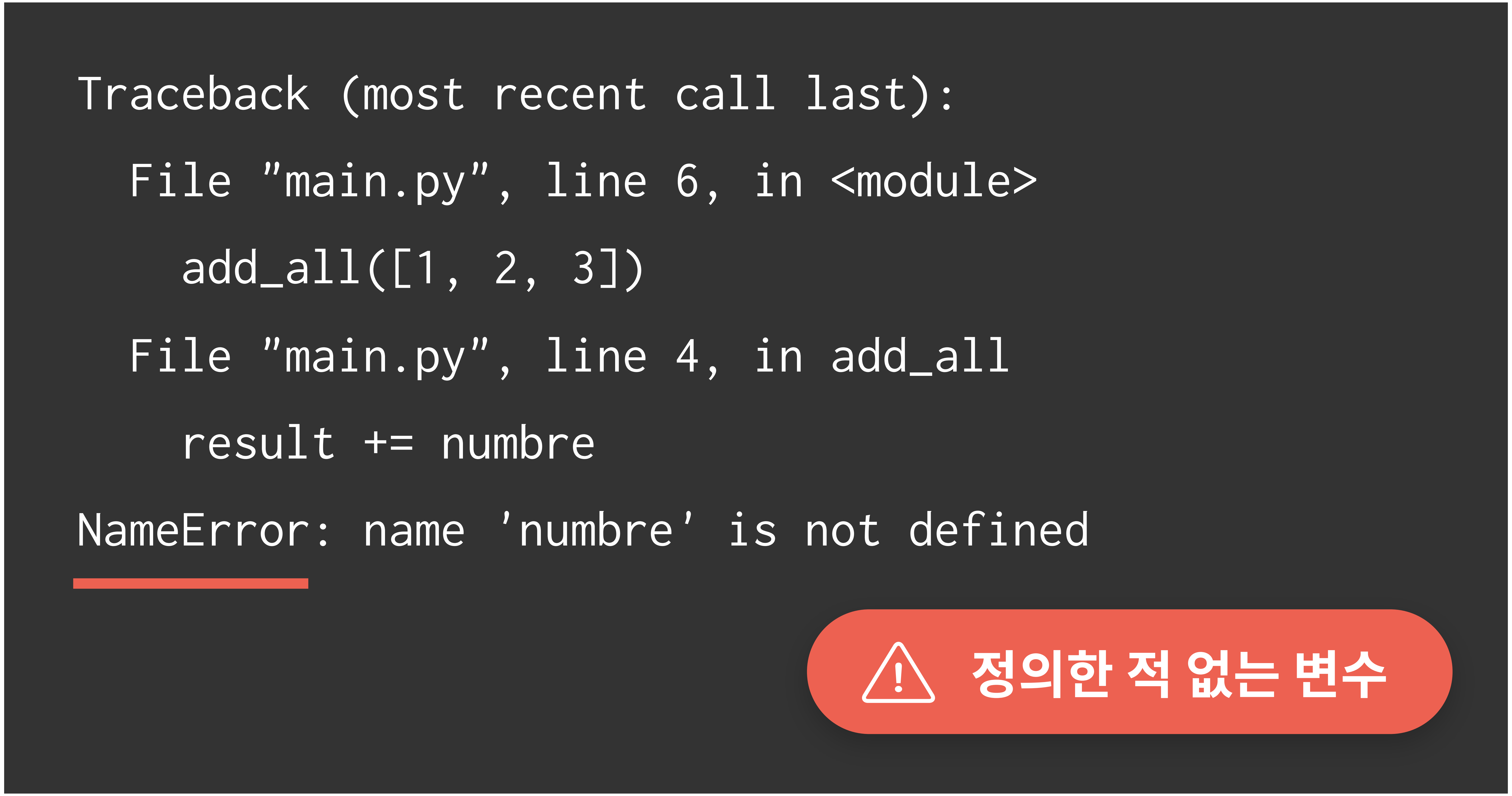
```
File ”main.py”, line 3
  for number in numbers

SyntaxError: invalid syntax
```

⚠ 컴퓨터가 이해할 수 없는 코드

Name error

```
def add_all(numbers):
    result = 0
    for number in numbers:
        result += numbre
    return result
add_all([1, 2, 3])
```

Name error



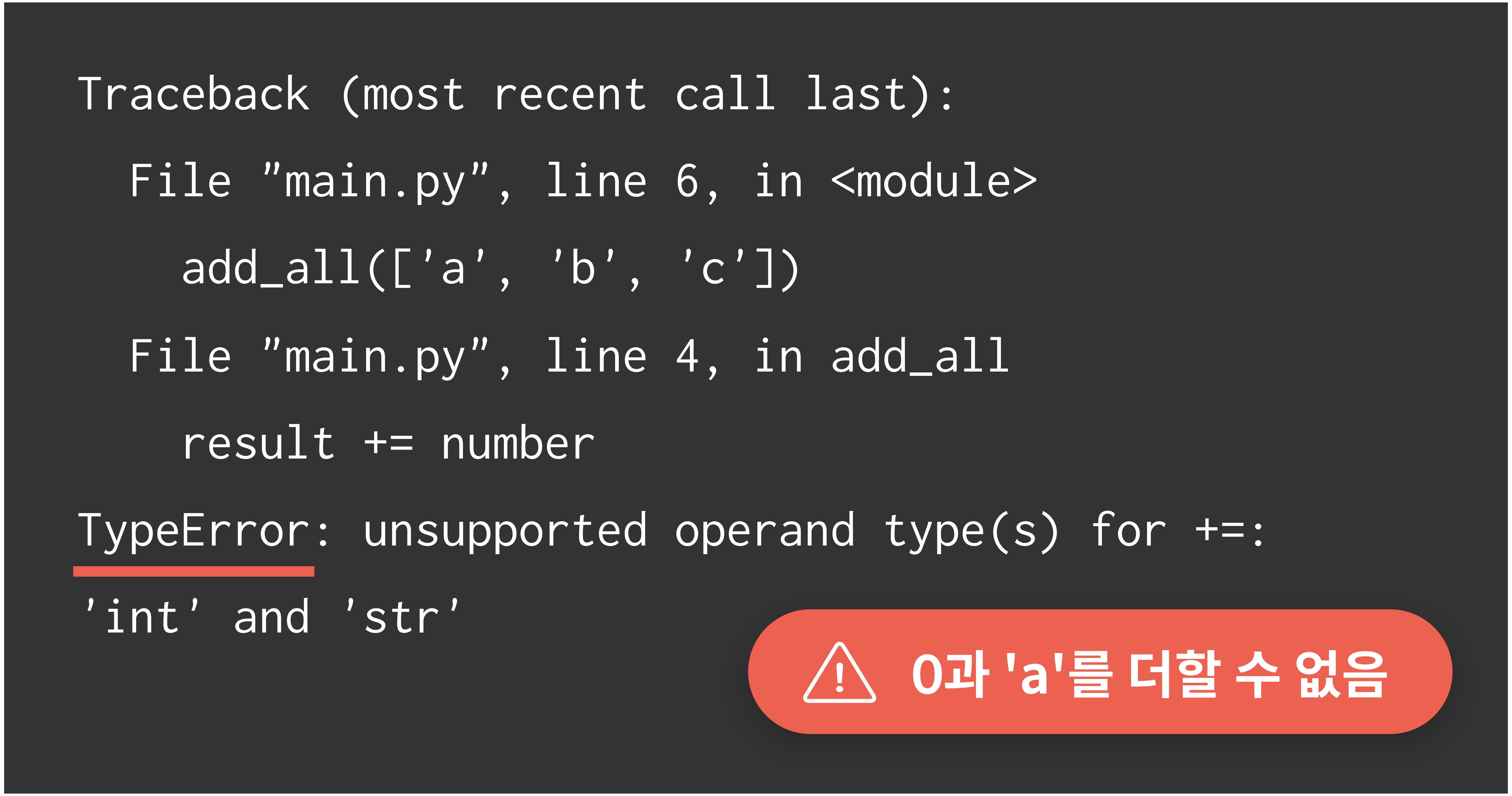
```
Traceback (most recent call last):
  File ”main.py”, line 6, in <module>
    add_all([1, 2, 3])
  File ”main.py”, line 4, in add_all
    result += numbre
NameError: name ’numbre’ is not defined
```

⚠ 정의한 적 없는 변수

Type error

```
def add_all(numbers):
    result = 0
    for number in numbers:
        result += numbre
    return result
add_all([’a’, ’b’, ’c’])
```

Type error



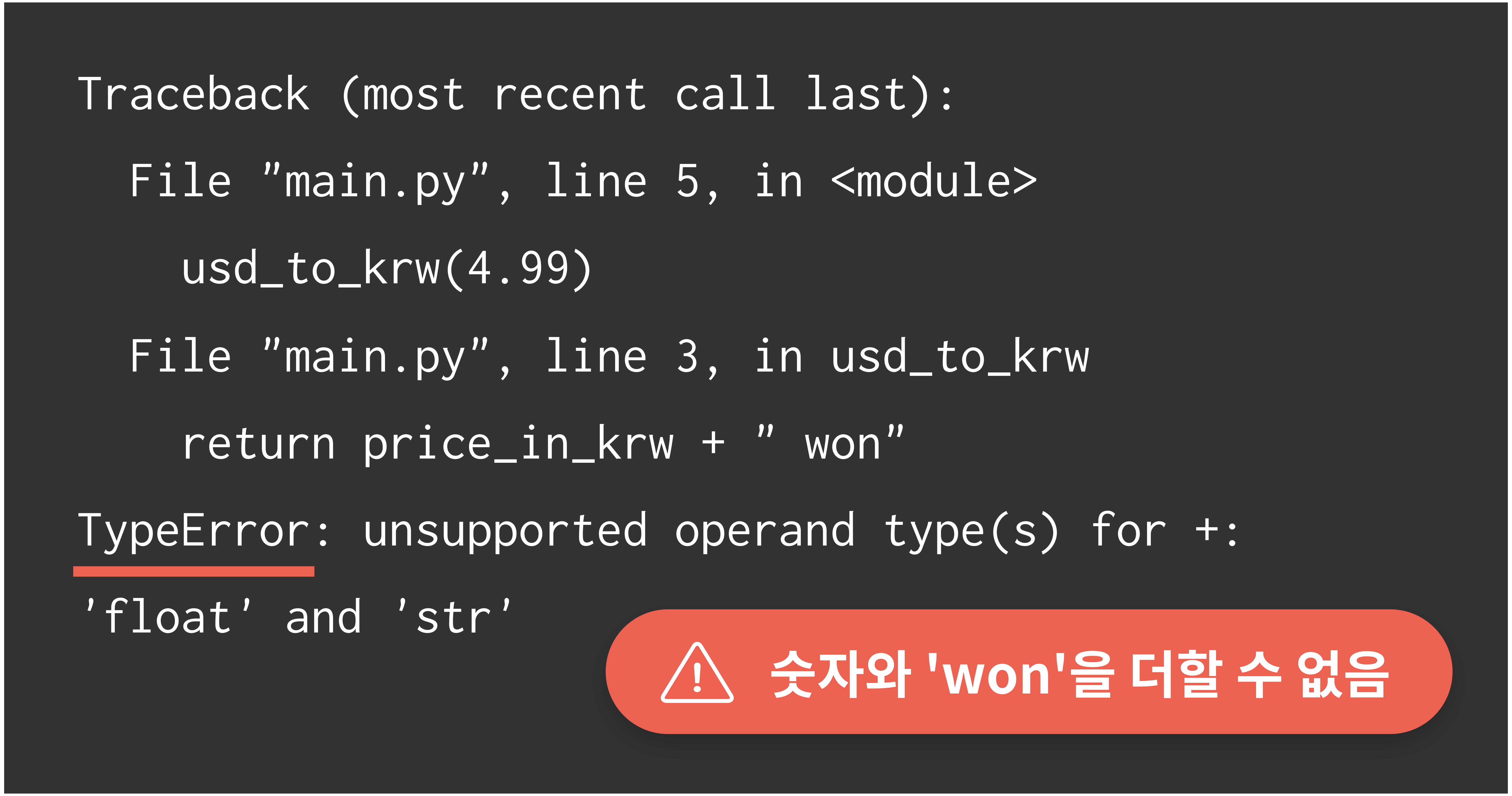
```
Traceback (most recent call last):
  File ”main.py”, line 6, in <module>
    add_all([’a’, ’b’, ’c’])
  File ”main.py”, line 4, in add_all
    result += number
TypeError: unsupported operand type(s) for +=:
’int’ and ’str’
```

⚠ 0과 'a'를 더할 수 없음

Type error

```
def usd_to_krw(price):
    price_in_krw = price * 1100
    return price_in_krw + ” won”

usd_to_krw(4.99)
```

Type error



```
Traceback (most recent call last):
  File ”main.py”, line 5, in <module>
    usd_to_krw(4.99)
  File ”main.py”, line 3, in usd_to_krw
    return price_in_krw + ” won”
TypeError: unsupported operand type(s) for +:
’float’ and ’str’
```

⚠ 숫자와 'won'을 더할 수 없음

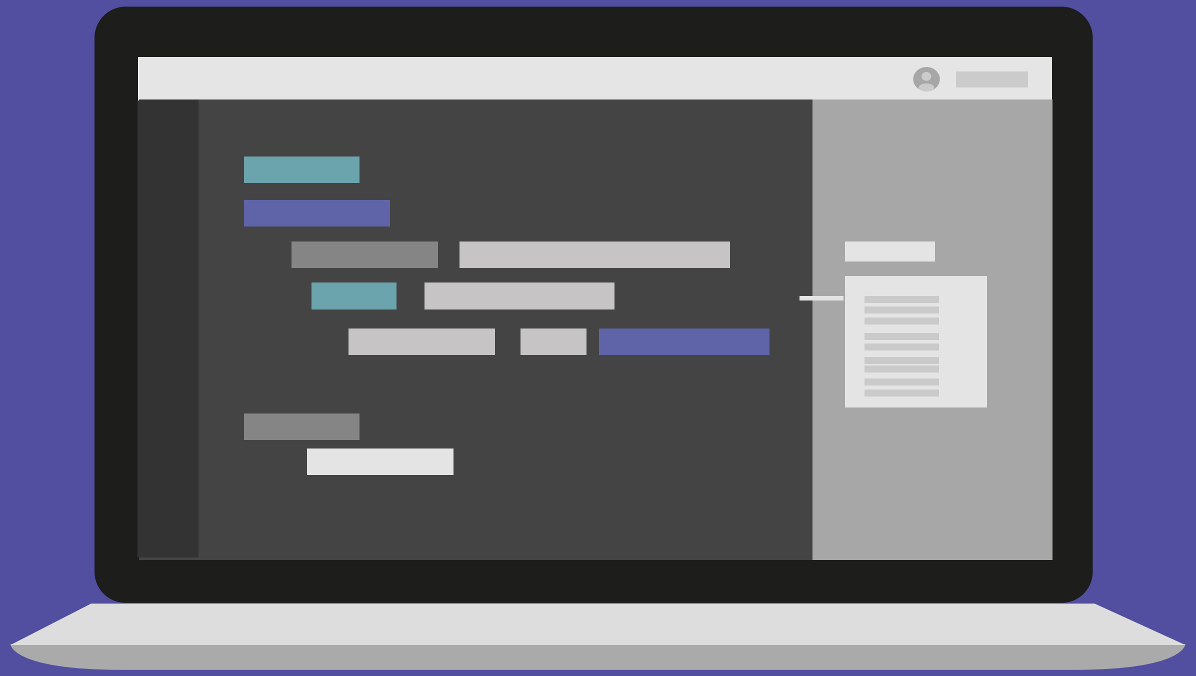
Type error

```
def usd_to_krw(price):
    price_in_krw = price * 1100
    return str(price_in_krw) + ” won”

usd_to_krw(4.99)
```

Type error

```
def usd_to_krw(price):
    price_in_krw = price * 1100
    return f”{price_in_krw} won”

usd_to_krw(4.99)
```

[실습 1] 성적 정리 1



Index error

```
def first_character(string):
    return string[0]

first_character(”hello”)    # h
```

Index error

```
def first_character(string):
    return string[0]

first_character(””)     # ?
```

Index error

```
Traceback (most recent call last):
  File "main.py", line 4, in <module>
    first_character("")
  File "main.py", line 2, in first_character
    return string[0]
IndexError: string index out of range
```

⚠ 문자열의 0번째 글자가 존재하지 않음

Index error

```
def first_character(string):
    if len(string) > 0:
        return string[0]
    return ””

first_character(””)    # ””
```

Index error

```
def first_character(string):
    if len(string) > 0:
        return string[0]
    print(”문자를 입력하세요!”)


first_character(“”)     # 문자를 입력하세요!
```

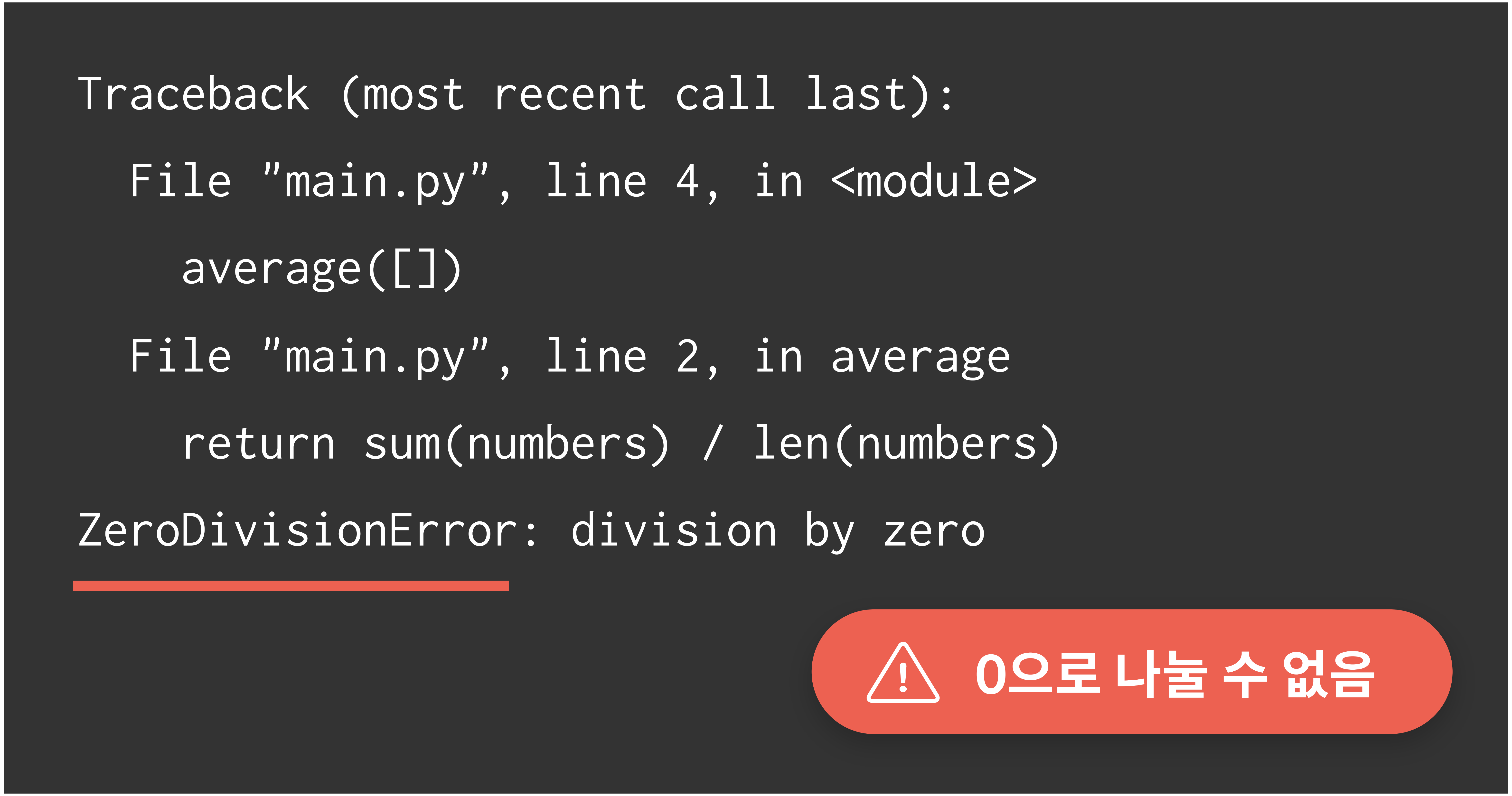
Zero division error

```
def average(numbers):
    return sum(numbers) / len(numbers)

average([60, 70, 80])    # 70
```

Zero division error

```python
def average(numbers):
    return sum(numbers) / len(numbers)

average([])    # ?
```

Zero division error



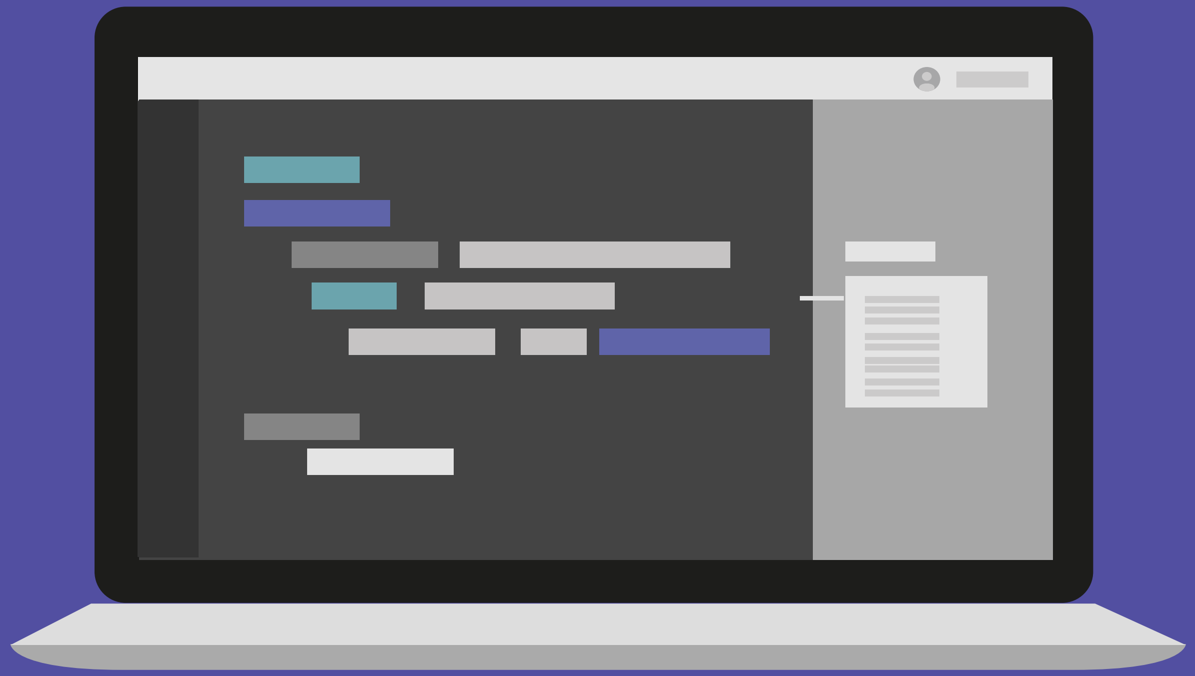
```
Traceback (most recent call last):
  File "main.py", line 4, in <module>
    average([])
  File "main.py", line 2, in average
    return sum(numbers) / len(numbers)
ZeroDivisionError: division by zero
```

⚠ 0으로 나눌 수 없음

Zero division error

```
def average(numbers):
    if len(numbers) > 0:
        return sum(numbers) / len(numbers)
    print(”No numbers!”)
```

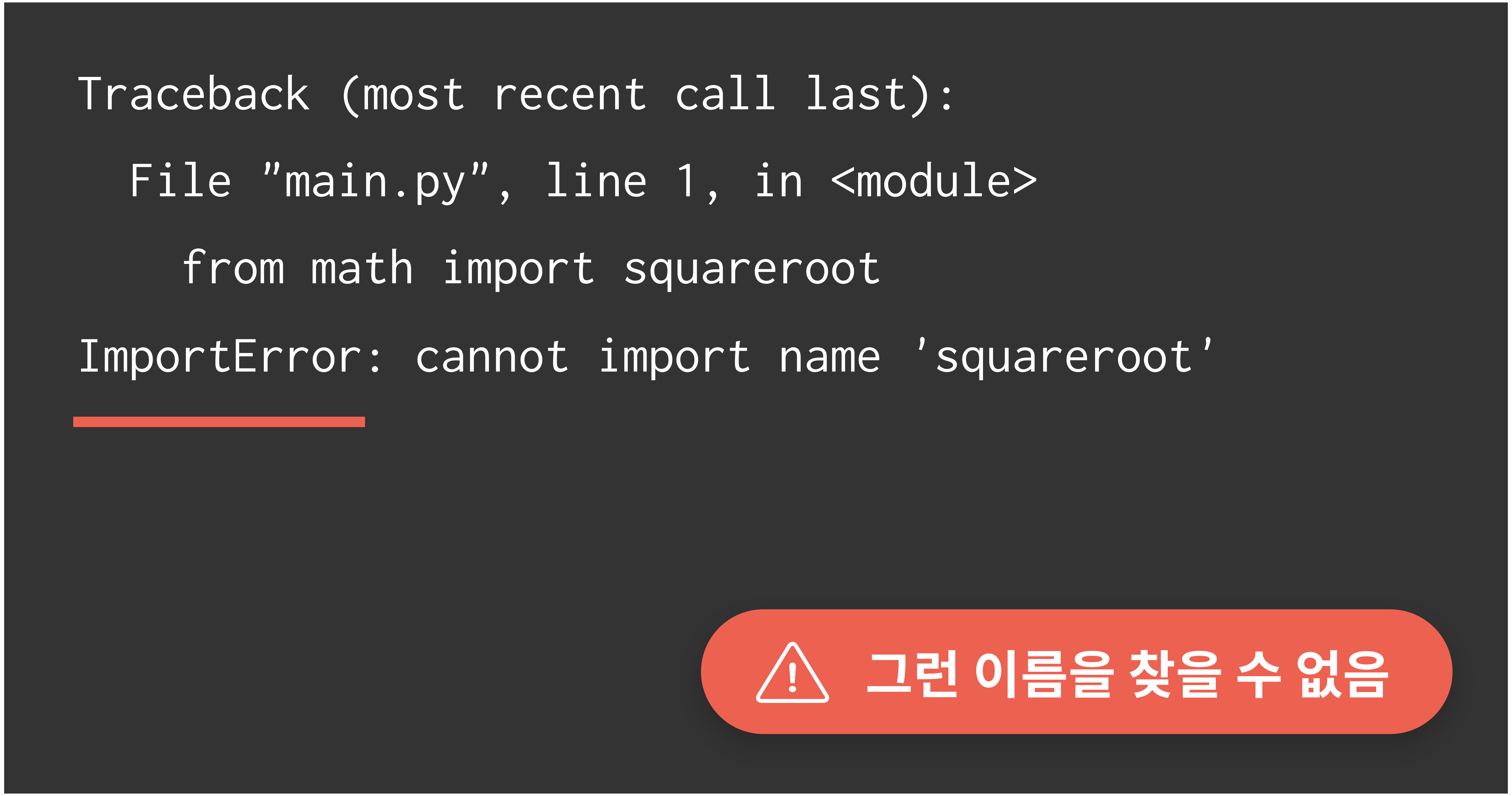
[실습 2] 성적 정리 2



Import error

```
from math import squareroot

print(squareroot(4))    # 2.0?
```

Import error



```
Traceback (most recent call last):
  File ”main.py”, line 1, in <module>
    from math import squareroot
ImportError: cannot import name ’squareroot’
```

⚠ 그런 이름을 찾을 수 없음

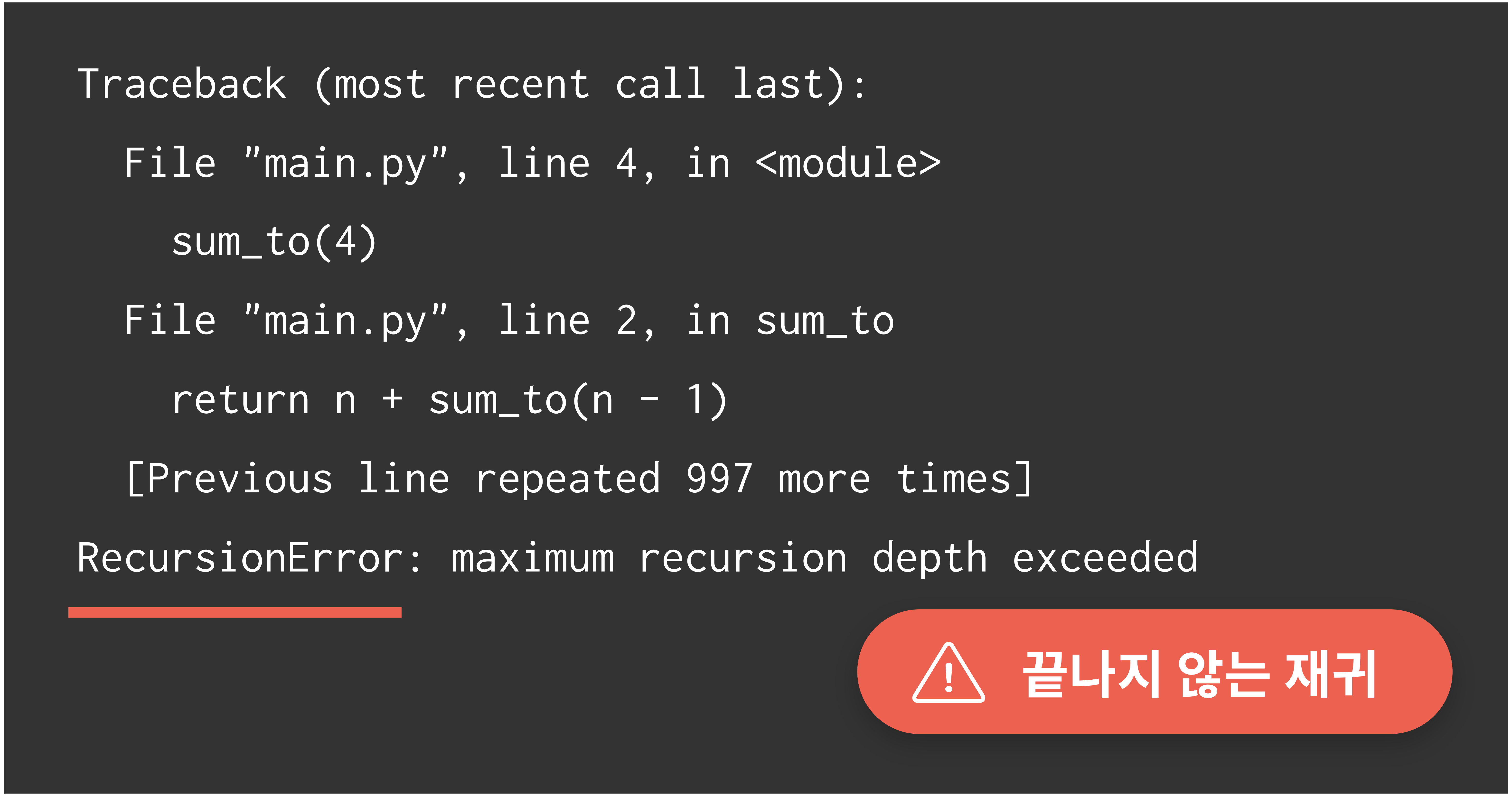
Import error

```
from math import sqrt

print(sqrt(4))     # 2.0?
```

Recursion error (심화)

```
def sum_to(n):
    return n + sum_to(n - 1)

sum_to(4)    # 10?
```

Recursion error (심화)



```
Traceback (most recent call last):
  File "main.py", line 4, in <module>
    sum_to(4)
  File "main.py", line 2, in sum_to
    return n + sum_to(n - 1)
  [Previous line repeated 997 more times]
RecursionError: maximum recursion depth exceeded
```

⚠ 끝나지 않는 재귀

Recursion error (심화)

```python
def sum_to(n):
    if n == 0:
        return 0
    return n + sum_to(n - 1)

sum_to(4)    # 10
```

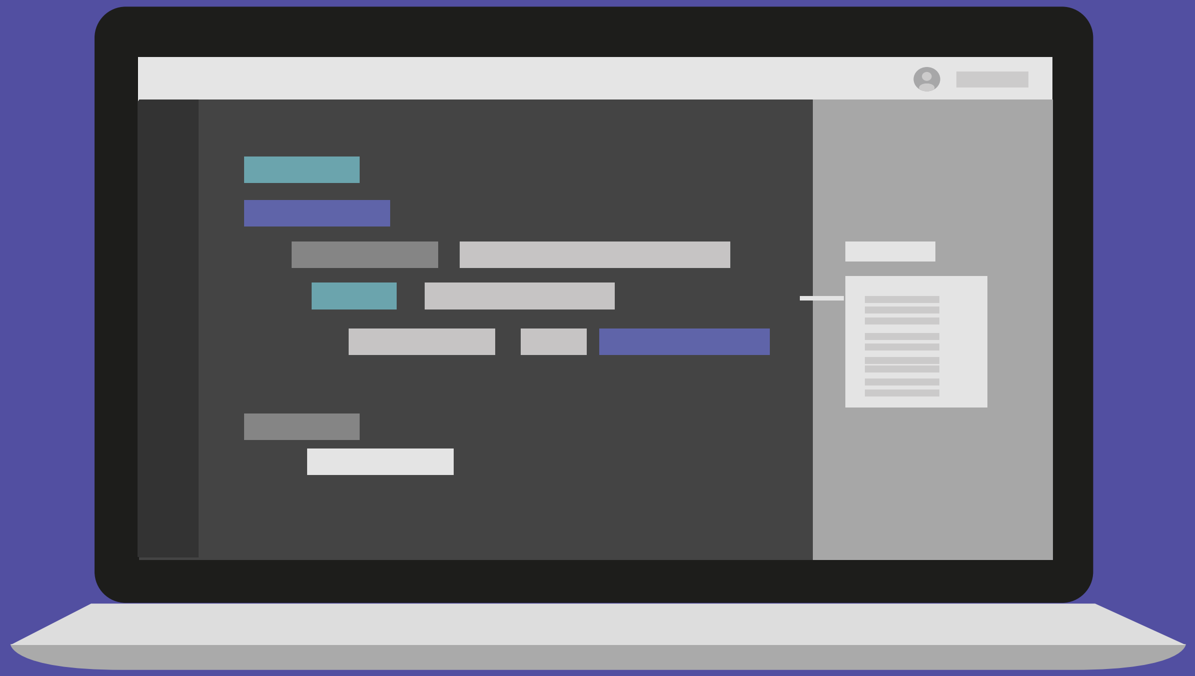
Recursion error (심화)

```
def sum_to(n):
    if n == 0:
        return 0
    return n + sum_to(n - 1)

sum_to(-1)    # ?
```

Recursion error (심화)

```
def sum_to(n):
    if n <= 0:
        return 0
    return n + sum_to(n - 1)

sum_to(-1)    # 0
```

[실습 3] 재귀 디버깅



try / except

에러 피하기

```
def average(numbers):
    return sum(numbers) / len(numbers)
```

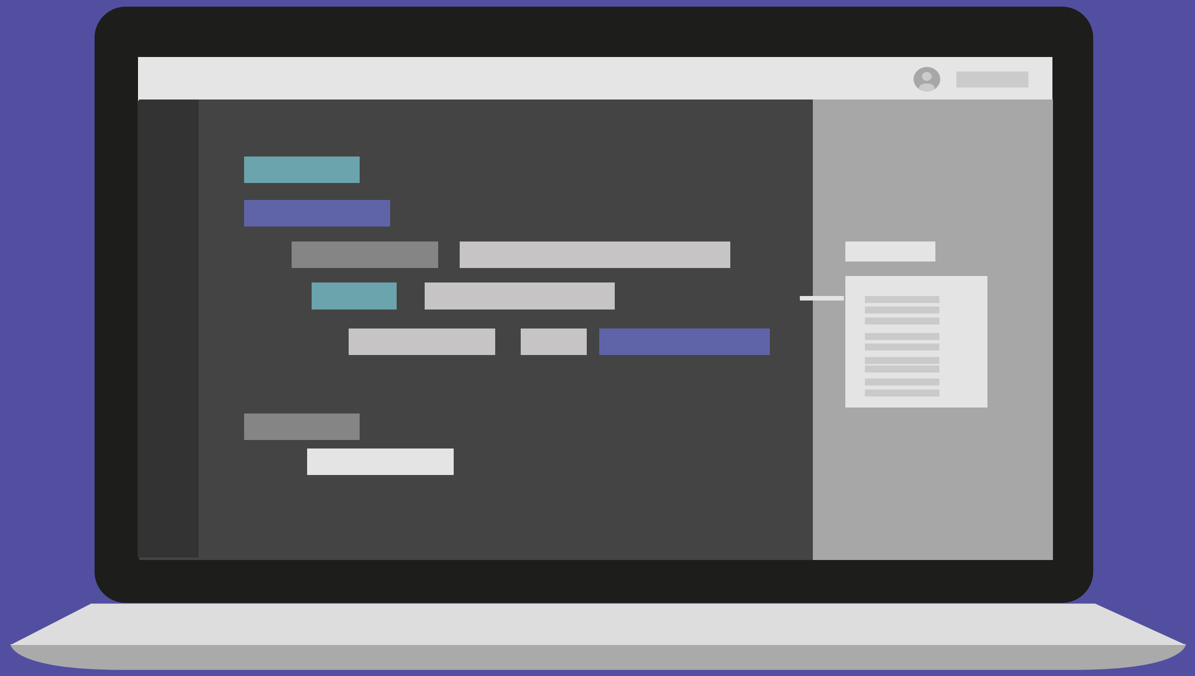
에러 피하기

```
def average(numbers):
    if len(numbers) > 0:
        return sum(numbers) / len(numbers)
    print(”No numbers!”)
```

에러 피하기

```
def average(numbers):
    try:
        return sum(numbers) / len(numbers)
    except ZeroDivisionError:
        print(”No numbers!”)
```

[실습 4] 성적 정리 3



/* elice */

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice