

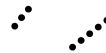
# 42강. finally 구문

혼자 공부하는 파이썬 - 윤인성 -

<https://www.youtube.com/playlist?list=PLBXuLglnP-5kr0PclHz1ubNZgESmliuB7>



# TABLE OF CONTENTS



1. finally 구문

# 1. finally 구문

**finally 구문**은 예외 처리 구문에서 가장 마지막에 사용할 수 있는 코드입니다. 예외가 발생하든 하지 않든 **무조건 실행할 때 사용하는 코드**입니다.

try:

예외가 발생할 수 있는 코드

except:

예외가 발생했을 때 실행할 코드

else:

예외가 발생하지 않았을 때 실행할 코드

finally:

무조건 실행할 코드

다음 슬라이드에서 예제를 살펴보겠습니다.

```

try:
    number = int(input("정수를 입력하세요> "))
    print("입력한 값은 {}입니다.".format(number))
except:
    print("정수를 입력하지 않았네요")
finally:
    print("정수를 입력해도 실행되고 하지 않아도 실행됩니다.")

```

```

PS C:\Users\User\Desktop\작업\Python> python hina.py
정수를 입력하세요> 1
입력한 값은 1입니다.
정수를 입력해도 실행되고 하지 않아도 실행됩니다.
PS C:\Users\User\Desktop\작업\Python> python hina.py
정수를 입력하세요> 싫어요
정수를 입력하지 않았네요
정수를 입력해도 실행되고 하지 않아도 실행됩니다.

```

finally를 안쓰고 그냥 맨 땅에 코드를  
입력해도 무조건 실행될텐데 왜 finally라는  
걸 만들었을까요..?

함수 내부 : return  
반복문 내부 : break

에서 finally를 사용할 때 의미를 갖게  
됩니다.

```
def test():
    print("test() 함수의 첫 줄입니다.")
    try:
        print("try구문이 실행되었습니다.")
        return
        print("try 구문의 return 키워드 뒤입니다.")
    except:
        print("except 구문이 실행되었습니다.")
    else:
        print("else 구문이 실행되었습니다.")
    finally:
        print("finally 구문이 실행되었습니다.")
    print("test() 함수의 마지막 줄입니다.")

test()
```

test() 함수의 첫 줄입니다.  
try구문이 실행되었습니다.  
finally 구문이 실행되었습니다.

try 구문 중간에서 탈출해도 finally 구문은 무조건 실행됩니다.  
따라서 함수 내부에서 **파일 처리 코드**를 깔끔하게 만들고 싶을때 finally 구문을 활용하는 경우가 많습니다. try구문에서 원할 때 return 키워드로 빠져 나가도 파일이 무조건 닫히기 때문입니다.

```

while True:
    print("test() 함수의 첫 줄입니다.")
    try:
        print("try구문이 실행되었습니다.")
        break
        print("try 구문의 return 키워드 뒤입니다.")
    except:
        print("except 구문이 실행되었습니다.")
        break
    else:
        print("else 구문이 실행되었습니다.")
    finally:
        print("finally 구문이 실행되었습니다.")
    print("test() 함수의 마지막 줄입니다.")

```

test() 함수의 첫 줄입니다.  
 try구문이 실행되었습니다.  
 finally 구문이 실행되었습니다.

반복문에서도 마찬가지로입니다. 반복문에서 break 키워드를 만나면 반복문을 빠져나가지만 finally 구문은 무조건 실행됩니다.

```
def write_text_file(filename, text):  
    try:  
        file = open(filename, "w")  
        return  
        file.write(text)  
    except Exception as e:  
        print(e)  
    finally:  
        file.close()  
  
write_text_file("test.txt", "안녕하세요~")
```

텍스트 파일을 만드는 함수인데 return 키워드 등으로 함수를 빠져나갈 때마다 close()를 하도록 코드를 작성했다면 코드가 굉장히 복잡해질 것입니다.  
하지만 이렇게 finally 구문에서 close() 함수를 호출하도록 코드를 작성하면 코드가 깔끔해집니다.

# 핵심 요약

파일 처리할 때 finally 키워드를 사용하면 좋다.

함수 return

반복문 break 등에서 쓸 수 있다.

〈try except 구문〉 - *try* 단독으로 못씀

try + except

try + except + else

try + except + finally

try + except + else + finally

try + finally

이 외의 조합은 실행할 때 구문 오류 발생.